

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平11-250279

(43)公開日 平成11年(1999)9月17日

(51)Int.Cl.<sup>6</sup>

G 0 6 T 15/00  
17/00

識別記号

F I

G 0 6 F 15/72  
15/62

4 5 0 A  
3 5 0 A

審査請求 未請求 請求項の数20 O L 外国語出願 (全 52 頁)

(21)出願番号 特願平10-375186

(22)出願日 平成10年(1998)11月20日

(31)優先権主張番号 0 8 / 9 7 5 1 3 3

(32)優先日 1997年11月20日

(33)優先権主張国 米国 (U S)

(71)出願人 599002892

リアル スリー・ディー

アメリカ合衆国、フロリダ州 32825、オ  
ーランド、レイク アンダーヒル ロード  
12506

(72)発明者 ハリー チャールズ リー

アメリカ合衆国、フロリダ州 32751、メ  
イトランド、レイモラッタ レーン 110

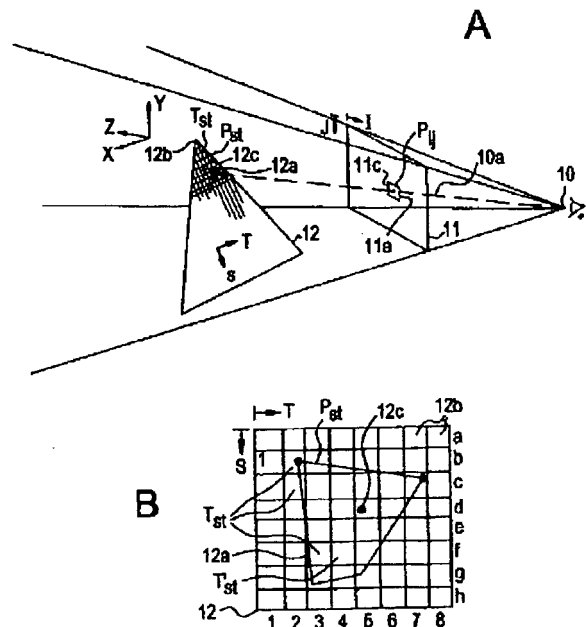
(74)代理人 弁理士 平田 忠雄

(54)【発明の名称】 コンピュータ画像形成ジェネレーションシステムにおけるシルエット/フットプリント解析を用いた異方性テクスチャマッピング

(57)【要約】

【課題】動作することが望ましくないときにロータをロックできる電気自動車制御装置を提供すること。

【解決手段】電気駆動自動車用ヒルホールド装置は、トルクコマンド信号TCMDを発生する「ガスペダル」を備えた車両を含んでいる。スイッチ(314)により、TCMDは、モータコントローラ(316、14)に結合される。モータコントローラ(316、14)は、モータ(40)を駆動し、したがって、自動車を駆動する。「ガスペダル」がトルクゼロを要求し、自動車速度がゼロのとき、スイッチは論理(図5)に応答し、オペレータ制御トルクコマンド信号TCMDを位置保持トルクコマンド信号 $T\theta$ に置き換える。位置制御トルクコマンド $T\theta$ は、ロータの角位置 $\theta$ を表す位置信号を受け取るコントローラ(312)により発生する。位置保持トルク制御ループは、次にモータのロータが指令位置から動かないようにするのに必要であるようなトルクを生成する。



**【特許請求の範囲】**

【請求項1】 コンピュータ画像ジェネレータにより表示すべきシーンにおける複数のピクセルの各々のテクスチャの属性を求める異方性マッピング方法であって、

(a) ピクセルごとに、そのピクセルのディスプレイスペースシルエットを、フットプリントとして、テクスチャスペースにおける少なくとも一つの関連対象ポリゴン上に射影する工程と、(b) 少なくとも一つのポリゴンを構成するテクセルの細部レベル(LOD)を調整して、射影ピクセルフットプリントが、テクスチャスペースの選択された方向における少なくとも所定数のテクセルをカバーするようにする工程と、(c) このフットプリントの総異方性領域をカバーする完全及び部分テクセルの累積数を求める工程と、(d) 前記ピクセルについて求めたビジュアル属性を、総累積テクセルから求める工程と、(e) 前記ピクセルについてのビジュアル属性情報を保存して、その関連シーンにおけるピクセルの表示を容易にする工程と、を含んでなることを特徴とする方法。

【請求項2】 工程(a)が、(a1) 少なくとも一つのポリゴンをカバーしているテクセル列にピクセルフットプリントを四角形として構成する副工程と、(a2) フットプリントの境界を定めるテクスチャスペースにボックスを構成する副工程と、(a3) 境界ボックス主方向を求める副工程とを含んでなる、請求項1に記載の方法。

【請求項3】 副工程(a3)は、ボックス主方向を、前記フットプリントの遠隔対向コーナー対と整列させる工程を含んでなる、請求項2に記載の方法。

【請求項4】 副工程(a2)は、平行四辺形によりフットプリントを近似する工程を含んでなる、請求項3に記載の方法。

【請求項5】 前記遠隔フットプリント対角線に最も近い平行四辺形対角線を、ボックス主方向として選択する、請求項4に記載の方法。

【請求項6】 工程(c)は、(c1) 一連の非標準化線方程式(UNLE)を構成して、テクスチャスペースにおけるフットプリントを特徴付ける副工程と、(c2) 上記一連のUNLEを使用して、前記境界ボックス内の各テクセルの中心方向を求める副工程と、(c3) フットプリントの合計に、前記一連の全てのUNLEに所定の関係を有する中心を有するテクセルのみを含める副工程とを含んでなる、請求項1に記載の方法。

【請求項7】 画像スペースのテクスチャスペースへの二次元変換用の一連の勾配を、UNLEの構成に使用する、請求項6に記載の方法。

【請求項8】 工程(a)が、(a1) 少なくとも一つのポリゴンをカバーしているテクセル列にピクセルフットプリントを四角形として構成する副工程と、(a2) フットプリントの境界を定めているテクスチャスペースに

ボックスを構成する副工程と、(a3) 境界ボックス主方向を求める副工程とを含んでなる、請求項7に記載の方法。

【請求項9】 副工程(a3)は、ボックス主方向を、前記フットプリントの遠隔対向コーナー対と整列させる工程を含んでなる、請求項8に記載の方法。

【請求項10】 副工程(a2)は、平行四辺形によりフットプリントを近似する工程を含んでなる、請求項9に記載の方法。

【請求項11】 前記遠隔フットプリント対角線に最も近い平行四辺形対角線を、ボックス主方向として選択する、請求項10に記載の方法。

【請求項12】 フットプリントの周囲についての予め選択された方向にUNLEを通す工程と、テクセルについてのUNLEの結果が同様に選択された結果の全てである場合のみ前記テクセルを総領域に含める工程とをさらに含んでなる、請求項6に記載の方法。

【請求項13】 各テクセルを、複数のサブテクセルに分割し、工程(b)～(d)の全てを各サブテクセルについて実施する、請求項1に記載の方法。

【請求項14】 工程(b)が、LODを選択して少なくとも5テクセルが選択された方向におけるピクセルフットプリントをカバーする工程をさらに含んでなる、請求項13に記載の方法。

【請求項15】 複数の細部レベル(LOD)に構成したテクセルデータを選択して画像スペースピクセルの表示に使用するためのコンピュータグラフィックス高速化方法であって、観測者に見えるオブジェクトによりシーンを定める工程であって、各オブジェクトが、テクスチャスペースに現れ且つテクセルの集団によりカバーされている少なくとも一つのポリゴンにより定められる、工程と、テクスチャオブジェクト上に射影したディスプレイピクセルシルエットのフットプリントを定める工程と、射影されたピクセルフットプリントの全領域をカバーする境界ボックスを定める工程と、テクスチャ用LODを選択して、所望数のテクセルを境界ボックスの主対角線に沿って配置させる工程と、前記フットプリント内にある選択された幾何学的位置を有し且つピクセルの表示可能属性に寄与する前記境界ボックス内の全てのテクセルの合計を求める工程と、を含んでなることを特徴とする方法。

【請求項16】 各テクセルを複数のサブテクセルにさらに分割し、そしてもしサブテクセルの選択された幾何学的位置がフットプリント内にあるならば、前記ピクセルの可視属性に寄与するサブピクセルを求める、工程をさらに含んでなる、請求項15に記載の方法。

【請求項17】 前記選択された幾何学的位置が、前記寄与セットに包含するかについて試験されているサブテクセルの中心である、請求項16に記載の方法。

【請求項18】 前記選択された幾何学的位置が、前記寄

与セットに包含するかについて試験されているテクセルの中心である、請求項15に記載の方法。

【請求項19】複数の非標準化線方程式を選択された位置と比較して、寄与セットに関連テクセルを含めることを決定する、請求項15に記載の方法。

【請求項20】コンピュータ画像ジェネレータにより表示すべきシーンにおける複数のピクセルの各々のテクスチャの属性を求める異方性マッピング用装置であって、次に処理される前記画像スペースピクセル(Pij)の指示を受けて、テクスチャスペースにおける少なくとも一つの関連対象ポリゴン上に関連ピクセルフットプリントをマッピングする手段と、テクスチャスペースの選択された方向における少なくとも所定数のテクセルによりカバーされるべき少なくとも一つのポリゴンをテクスチャリングするテクセルの細部レベル(LOD)を調整する手段と、このフットプリントの総異方性領域をカバーする完全及び部分テクセルの累積数を求める手段と、前記ピクセルについて求めたビジュアル属性を、総累積テクセルから求める手段と、前記ピクセルについてのビジュアル属性情報を保存して、その関連シーンにおけるピクセルの表示を容易にする手段と、を含んでなることを特徴とする装置。

#### 【発明の詳細な説明】

【発明の属する技術分野】本発明は、三次元(3D)シーンの二次元(2D)表現の表示に用いられるコンピュータ画像形成(CIG)システム(グラフィックスアクセラレータとも称される)に関し、より詳細には、効率的な計算方法で、CIGディスプレイ上に各(好ましくはリアルタイム)シーンを形成する複数のポリゴンの各々について必要とされる各表示スペースピクセルのシルエット解析により、テクスチャを各表示ピクセルに異方的にマッピングする新規な方法及び装置に関する。

【従来の技術】現在では、2Dディスプレイ上に3D画像シーン(各シーンは、何千(典型的には100万以上)もの個々の画素(ピクセルと称される)を含む)があり、各ピクセルは、3Dオブジェクトスペースの2D画像スペース表示における対応の小さな可視領域に特有のビジュアル属性を有し、したがってオブジェクトに対して一定の向きに位置にしている観察者が観察できる)を形成するのに、コンピュータ画像形成システム及び関連プロセスを提供することは周知である。オブジェクト自体は、既知の観察点から見られ、したがって、一連の平面ポリゴンにより表すことができる。平面ポリゴンは、テクスチャスペースへの射影によりテクスチャを形成して、選択されたテクスチャパターンのテクスチャ要素(テクセル)で重畳してから、表示を行う。CIGテクスチャードポリゴンプロセスの種々の態様については、米国特許第4,727,365号(Advanced Video Object Generator、

1988年2月28日発行)及び米国特許第4,974,176号(Microtexture for Close-In Detail、1990年11月27日発行)を含む入手可能な明細書及び特許を参照できる。上記両方の特許は、本出願の譲受人に譲受され、そこに引用されている文献の全てとともに、引用することによりそれらの全体が本発明の開示の一部とされる。リアルタイムで使用され且つインタラクティブグラフィックシステムを用いたほとんどのテクスチャマッピング法では、視角(オブジェクトポリゴン面に対して)が小さいテクスチャードオブジェクトでは、即ち、テクスチャリングされるべきオブジェクトが高アスペクト比を有する(即ち、ポリゴン法線に対しての視角が大きく、一般的に45°超である)場合には、上質のテクスチャマッピングができないことが分かった。この一例として、「ホワイトラインダウンザセンターオブザロード(White Line Down the Center of the Road)」の問題がある。現実の世界では、断続白色路面ラインは、一連の白色ラインセグメントから構成されている。これらの白色ラインセグメントは、各々長さ約10フィート、幅5インチであり、各白色ラインセグメントは10フィートの黒色舗装により分離されている。リアル3D白色路面ラインの2D表現の表示では、2つの相反する目的を達成しようとする試みがなされるであろう。即ち、第一の目的は、観察者が路面を見下ろしたときに白色ラインが見えなくなるのを防止することであり、第二の目的は、得られた表示白色ラインのちらつきを防止することである。ここで、ちらつきは、路面に沿った種々のセット(例えば、長さ50フィート)のいくつかの白色ラインセグメントが出現したり消えたりして現れる。これらの問題の別の関連した例として、盛り上がり凹凸の変化がある。即ち、観察者と対向しているときには鮮鋭であるが、高アスペクト比(観察者に対する角度が小さい)で見ると、ちらつつか、ソフトな外観となる。これらの問題のほとんどは、テクスチャスペースにマッピングしたときに、細長いフットプリントを有する表示ピクセル(画像スペースにおいて、通常は長方形であり、正方形、シルエットのこともある)とならないことにより生じるものである。テクスチャマップは、しばしば、細部レベル(LOD)セットに構成される。典型的には、セットの各々のより粗いレベルの要素は、それよりも次に最も微細なセット要素に対して2:1の割合で減少するテクスチャマップである。即ち、レベルDのテクスチャマップは、次に粗いレベルD+1についてのテクスチャマップにおける各テクセルごとに4(又は2x2)のテクスチャ要素(テクセル)を有する。したがって、最も微細な第一LOD(D=0)マップが512x512テクセル列を有するならば、次に粗いLOD(D=1)は、256x256テクセル列から構成されるテクスチャ

チャマップである。このプロセスは、例えば、LOD 9 が、単一の  $1 \times 1$  テクセルを有するマップを生じるまで実施される。従来の通常のテクスチャピクセルマッピングでは、各ピクセルの有効 LOD は、テクスチャスペースに射影されるピクセルの幅と長さの両方について計算され、これら 2 つの有効 LOD 数の大きいものを、そのピクセルの有効 LOD として選択する。次に、関係のあるピクセルの有効 LOD を同類として組分けした LOD を有する 2 つのテクスチャマップを選択し、3 つの補間法のうちの一つを使用して、2 つの隣接 LOD テクスチャマップ間の補間を行って、マッピングしたピクセルの値を算出する。3 つの補間法とは、ニアレストネイバー補間、双一次補間又は三線補間である。テクスチャスペースに射影したときの修正ピクセルの幅又は長さのうちの大きい方を使用してピクセルの有効テクスチャスペース LOD を求めるので、射影ピクセル有効サイズを、その長さ方向と幅方向の両方において同じテクスチャスペースサイズとなるようにする。これらのテクスチャ法は、マッピングしたピクセルの LOD は常にテクスチャスペースにおいて両方向で同じであるので、等方性であると言われている。しかしながら、一般的に長方形（好ましくは、正方形）の画像スペースピクセルを、視軸（自体画像／表示スペース面に直交している）に対してある程度斜めになっているオブジェクトスペースポリゴン上に射影すると、射影角が小さいので、各射影又は（テクスチャ）マッピングしたピクセルは、細長くなる。このような伸び（大きい射影寸法の小さい射影寸法に対する比）が、ある選択された限界（例えば、3 : 1）を超えると、長さが幅に等しいとする仮定を用いた周知の等方法に適用できない。これは、射影ピクセルフットプリント寸法及び、したがって、関連 LOD を、両方向において同じにすると望ましくないビジュアルアーチファクトを生じてしまう、異方性の場合である。テクスチャスペースにおいて射影ピクセル細長形状を形成する異方テクスチャ法を提供することが、非常に望ましい。いくつかのこのような方法（通常、非リアルタイムシステム）が、大きく分けた 2 つの種類のうちの一方向に入る手法により提供された：1）テクスチャ値を用いてピクセルテクスチャスペース射影を重畳する方法。非常に正確な結果が得られるが、計算に要する費用が非常に高い。2）予備処理及び予備フィルター処理したテクスチャードマップの保存を用いた方法。この方法では、計算はさほど必要ないが、融通がきかず且つ精度が低く、さらにはメモリーをくう。リアルタイム且つインタラクティブなシステムにますます異なるテクスチャマップを利用することが要求されてきているので、メモリー使用量の少ない手法がより望ましい。従来、異方性テクスチャを提供する他のスキーム（我々の前任者のシュミュレーショングループによりほぼ 10 年前に使用された、異方性ピクセルの単一のより大きな対角線のみに沿ってテ

クスチャをサンプリングする方法等）が提案されたが、これらの方法は、いままでに、異方性テクスチャリング問題を解決するには不十分であるか、コストが高すぎる（必要なハードウェアや計算時間の面で）ことが判明した。したがって、効率的な計算方法で、CIG ディスプレイ上に各（好ましくはリアルタイム）シーンを形成する複数のポリゴンの各々について必要とされる各表示ピクセルに、テクスチャを異方的にマッピングする新規な方法を提供することが、非常に望ましい。

【発明が解決しようとする課題及び課題を解決するための手段】効率的な計算方法で、CIG ディスプレイ上に各シーンを形成する複数のポリゴンの各々について必要とされる各表示ピクセルに、テクスチャ情報を異方的にマッピングする本発明による方法では、テクスチャアレイにマッピングした各変換ピクセルのフットプリントを使用して、完成テクセル及びテクセル部分がピクセルフットプリントにてカバーされる関連長さ及び関連幅の両方を有する異方性テクスチャスペース領域を決定する。特定のピクセルフットプリントを特徴付ける一連の非標準化線方程式を使用して、前記ピクセルフットプリント付近の各ピクセルが、前記射影ピクセルフットプリントにより包囲されるか、回避される（即ち、その内側か外側）を決定する。1D ライン（極めて高いアスペクト比（典型的には  $70^\circ$  を超える）の場合にのみ適当である）ではなく等方性 2D 領域を使用することにより、この方法では、含まれる全テクセル又はテクセル部分の計算を考慮され、ピクセルのカラー及び／又は半透明性を求められる。全ての値がインクレメント法で求められるので、本発明の方法では、最初にいくつかの値の計算をしてから、単純インクレメント法によってどのテクセルがフットプリントに含まれ、どれが含まれないかを計算する必要がある。テクセルをさらに 2 以上のサブテクセルに分割するならば、各サブテクセルについても、ピクセルフットプリントの中から外かを求めることができるので、テクスチャマッピングがより正確になる。ピクセルは、その有効等方性 LOD が示すよりも高い解像度のテクスチャマップにマッピングされる。射影ピクセルは、全方向において多数のテクセルと重なる。したがって、本発明の一つの目的は、面射影及びシルエット解析を用いた画像スペースピクセルの異方性テクスチャリングのための新規な方法及び装置を提供することである。本発明のこの及び他の目的は、添付図面を考慮し且つ現在のところ好ましい実施態様の関連した詳細な説明を読むことにより、当業者には明らかとなろう。

【発明の実施の形態】まず、図 1 a において、観測者 O は、視点 10 で、軸 I 及び J により特徴付けられる画像又は表示、スペースである表示面 11 を見る。この面は、表示画素列、即ち、ピクセル列から構成されている。一般的なアレイピクセル  $P_{ij}$  の 11 a は、それぞれの I 及び J 軸寸法により表示される。視線 10 a

は、ピクセル11aシルエットの中心11cを通過し、テクスチャスペースにおける面12上に射影すると、ピクセル11aのフットプリント12aの中心12cとして射影される。面12は、一般的に、テクスチャスペースに射影するとオブジェクトスペースポリゴンであり、その上にテクスチャ要素、即ち、テクセル12b列を有する。テクスチャードポリゴン面12は、軸S及びTを有するので、一般的なアレイテクセルTSTがそのS及びT軸寸法により表示される。ピクセル11aの一般的に長方形である（正方形のことがある）シルエットは、その下にあるポリゴン12が一般的にスクリーン面11に平行ではないので、四辺形ピクセルPSTフットプリント12a（典型的には4辺が等しくないだけでなく、コーナーが等しくなく典型的には直角でない）として面12上に射影されることが分かるであろう。テクスチャリング問題は、図1bを参照することにより理解されるであろう。図1bにおいて、テクセル面12は、読み取り者の視線に直交しているので、アレイのテクセル12bは、長方形、好ましくは正方形、であるが、射影ピクセルPSTフットプリント12aは、ゆがんだ四辺形である。テクスチャ細部レベル（LOD）を選択して、各テクセルの大きさが一般的に射影ピクセルPSTの大きさよりも小さいようにした。したがって、ピクセル12aは、少なくとも一つのテクセルTSTを部分的にカバーし、且つ一つ以上の他のテクセルT'STを完全にカバーしてもよい。ここで、S方向におけるテクセル列は、a～hと表示し、一方、T方向におけるテクセル列は、1～8と表示した。フットプリント12aは、テクセル3c、4c、3d、4d、5d、4e、5e及び4fを完全にカバーするが、各テクセルが特有の一連のビジュアル属性、例えば、カラー／透明性値、を有することができる（おそらく）テクセル2b、3b、4b、5b、2c、7c、3e、6e、3f、5f、6f、3g、4g及び5gは部分的にカバーする。当業者により認識されているように、問題は、特定のピクセルPSTフットプリント12aの総領域により部分的に包囲されているか、完全に包囲されている全てのテクセルの寄与を適切として、ピクセルPijの表示スペース部位での表示のための適切なカラー、半透明性等の可視属性を求めることである。図2において、本発明の新規な方法の一つの好ましい実施態様において、3Dコンピュータ画像ジェネレータ（CIG）は、流れ図20に示す等方性テクスチャリングシルエット／フットプリント解析法を実施するグラフィックスプロセッサ（専用ハードウェア論理エンジンでも、ソフトウェアプログラム化汎用コンピュータでもよい）を備えたものでよい。プロセスは、ステップ21から開始する。ステップ23では、テクスチャリングされるべき次のポリゴン12を選択する。この実施態様では、ピクセルを、グラフィックエンジンにより提供される各特定のポリゴンごとにグループ化して

よいが、他のピクセル操作法も同様に利用できる。ステップ25では、手順20により、現在選択されているポリゴンNにおいてテクスチャリングされるべきである次のピクセルPijの1対の値I及びJを選択する。この選択により、テクスチャスペースにマッピングしたときのピクセルPijコーナーの画像スペース座標（i、j）とピクセルPstのコーナー座標（s、t）の両方が効果的に求められる。また、図2aにおいて、選択された4コーナー四辺形ピクセル11aは、一定のコーナーp1～p4を有している。コーナーの各々は、その関連I、J座標を有する。即ち、p1は（i1、j1）、p2は（i2、j2）、p3は（i3、j3）、p4は（i4、j4）にある。表示ピクセルPijシルエット11aは、テクセルスペースピクセルPSTフットプリント12aに変換され、画像スペースコーナーPijの各々は、それぞれ関連テクスチャスペースコーナーPSTに変換される。即ち、p1は（s1、t1）、p2は（s2、t2）、p3は（s3、t3）、p4は（s4、t4）にある。ステップ27では、ピクセルテクセル空間勾配を求めて、ピクセルフットプリントをステップ29におけるテクセルアレイに構成される。図2bに示すように、スクリーンスペースからテクスチャスペースに、マッピング勾配が見られる。部分空間デリバティブ $\partial s / \partial i$ 及び $\partial t / \partial i$ が、第一コーナー（s1、t1）について見られ、部分空間デリバティブ $\partial s / \partial j$ 及び $\partial t / \partial j$ が、第二コーナー（s2、t2）について見られる。これらの空間デリバティブを所与のものとすると、各ピクセルp（i、j）の（i、j）座標は、テクスチャスペースにp（s、t）、即ち、p（i1、j1）→p（s1、t1）；p（i2、j2）→p（s2、t2）；p（i3、j3）→p（s3、t3）；及びp（i4、j4）→p（s4、t4）としてマッピングできる（マッピングは符号→により示されている）。第一コーナー（s1、t1）が、多少長い射影変換プロセスにより判明する場合には、他の3つのコーナーは、画像スペースにおける側長及びマッピング勾配の情報から分かる。通常の場合では、各ピクセルが関連勾配項の乗数としての役割を果たす単位側長を有する場合、これらの他のコーナー位置は以下の通りである：

$$(s2, t2) = (s1 + \partial s / \partial j, t1 + \partial t / \partial j),$$

$$(s3, t3) = (s2 + \partial s / \partial j, t2 + \partial t / \partial j) \text{ 及び}$$

$$(s4, t4) = (s1 + \partial s / \partial j, t1 + \partial t / \partial j).$$

完全なテクスチャスペースピクセル射影フットプリント12aは、これらの4つのコーナーの規則的相互接続により構成できる。また、図2cにおいて、ステップ31に入って、境界BBボックス14を演算する。この境界BBボックス14は、フットプリントの最大、最小S値

及び最大、最小T値で、コーナー点Pの一つにちょうど触れている長方形ボックスであり、したがって、ピクセルフットプリント12aを包囲する。次に、ステップ33において、最初のフットプリント12aの主対角線12dを形成する最遠隔対向コーナー（ここでは、対向コーナーp2及びp4よりも離れているコーナーp1及びp3）と関連且つ最近接しているボックス14の対角線的に対向するコーナー対（ここでは、コーナーp'1及びp'3）間の境界ボックスの主対角線14aを計算する。対角線14aは、一般的にフットプリント主対角線12dと一致するが同一ではないことが分かるであろう。境界ボックス14は、S及びT方向の各々におけるフットプリントの最大及び最小サイズにより決定されるので、ステップ31のプロセスは具体的に以下の通りであることができる：

$s_{\max} = \max(s_1, s_2, s_3, s_4)$  及び  
 $s_{\min} = \min(s_1, s_2, s_3, s_4)$ 、

一方

$t_{\max} = \max(t_1, t_2, t_3, t_4)$  及び  
 $t_{\min} = \min(t_1, t_2, t_3, t_4)$ 。

式中、 $\max$  及び  $\min$  は、それぞれ以下の括弧内の量の最大又は最小のものを示す。ここで、 $s_{\max} = s_4$  及び  $s_{\min} = s_1$  であり、一方、 $t_{\max} = t_3$  及び  $t_{\min} = t_1$  であることが分かる。上記から、グラフィックプロセッサは、次に、ボックスS範囲  $R_s = (s_{\max} - s_{\min})$  又は  $(s_4 - s_1)$  及びボックスT範囲  $R_t = (t_{\max} - t_{\min})$  又は  $(t_3 - t_1)$  を計算できる。境界ボックスの主方向は、工程33において、Sサイズ及びTサイズの最大値を比較することにより決定される。範囲  $R_s$  と範囲  $R_t$  のうちの大きい方により、主方向DMが設定される。即ち、 $D_{\max} = \max(R_s, R_t)$  であり、 $D_{\max} = R_s$  の場合（即ち、 $R_s > R_t$  の場合）、 $DM = R_s$  であり、 $D_{\max} = R_t$  の場合（即ち、 $R_t > R_s$  の場合）、 $DM = R_t$  である。次に、ステップ35に入り、適当なLOD

「D」を求める。計算を簡略にするためには、もし複数のテクセル12bが各射影ピクセルフットプリント12aをカバーするならば、四辺形フットプリント12aを最小可視誤差を有する適当なサイズの平行四辺形12Aにより近似できることが分かった。したがって、ピクセルフットプリント境界ボックス14領域が少なくとも一つのボックス方向においていくつかのテクセルをカバーするのがよい（図2d）。両軸方向に複数のテクセルをカバーするのが好ましいが、高アスペクト比を有する射影を除外し、一つの軸方向に沿ってのみ複数のテクセルをカバーしてもよい。本発明の面異方法では、少なくとも一つのテクセルを有し、好ましくは、「副」方向であってもテクセル数は多いほうがよく、ラインに沿ったのみのサンプリングカベレージではなく真のカベレージ領域を有するのがよい。もし高アスペクト比フットプリントを、通

常のように演算するならば、以下で説明するサブテクセル状態を通常使用できる。これにより、テクセルの一部として測定されるサイズで第二寸法を使用でき、極めて小さい寸法（約90°で、ポリゴン面エッジオンで見られることのあるサブテクセル未満）を有するボックス14を無視できる。ボックスのS軸側14sによりスパンニングされているテクセル12bの数TSを計算し、ボックス14のT軸側によりスパンニングされているテクセルの計算で求めた数Ttと比較する。LOD数Dを変更して、TS又はTt付近の所望の数とすることができる。Dの良値がわかったら、LODに適当なテクスチャグリッドを、残りの工程に使用されるメモリーから抽出する。テクスチャグリッド配置は、グリッドLODが奇数か偶数か（ピクセルをスパンニングしている最大テクセル数の目安）に依存し、最微細解像度LODのDは、条件  $(TD/2D) \leq D_{\max}$  が満たされる限り、テクセルグリッドの一方においてカバーされているテクセルの数TDを検討することにより設定される。典型的には、3〜5テクセルによりS範囲及びT範囲のうちの長い方に沿ってピクセルをスパンニングすることを試みた。ステップ37（再び、図2d参照）において、本発明の方法では、テクセル12bがフットプリントの中にあるか（したがって、ピクセル12aにおけるそのカラー及び半透明性属性を有する）、フットプリントの外にあるか（及び増分ピクセル属性に加えない）を決定する。これには、粗マッピングを得るための一連の4つの非標準化線方程式（UNLE）L1〜L4を利用して、テクセルの包含について決定する。ラインL1〜L4の各々は、フットプリント12aに関してボックス14内への各テクセルの包含又は除外を決定するのに使用される自体のUNLEによって表される。全体（完全に包囲）及び部分テクセル12bの両方の寄与を合計して、ピクセルフットプリント属性を形成する。非標準化線方程式を使用していずれかの点の位置を決定することと、したがって、テクセルがピクセルフットプリントの一部分でなければならないかどうかは、どの側のラインで調査中の点（例えば、各テクセルの中心12b'）がオンであるかをUNLE決定することにより行う。周囲点12a'が、常に一貫した方向（例えば、時計方向）におけるサイドラインLに沿ってフットプリント付近を移動するならば、既知の方向（例えば、ラインLの右）におけるサイドラインLからずれている点12b'は、ピクセル内にある可能性があり、4つのサイドラインL1〜L4全てについて点12b'を試験して、包含又は除外を決定する。したがって、UNLEプロセスにおける第一ステップでは、4つの非標準化線方程式を生成する。本発明によれば、UNLEが、 $L = L_o + (L_s \times \Delta s) + (L_t \times \Delta t)$  の形態にあるのが好ましい。例えば、第一UNLE、一定の最終点  $(s_1, t_1)$  及び  $(s_2, t_2)$ 、の係数は、 $L_o = (s_2 \times t_1) -$

( $s_1 \times t_2$ )、 $L_s = (t_2 - t_1)$  及び  $L_t = (s_1 - s_2)$  である。したがって、4つのUNLEの各々は、ピクセルフットプリント12aのエッジである。各テクセル12bを試験して、それが中にあるか外にあるかを決定できる。この試験は、もしテクセルスペースにおけるピクセルのフットプリントが、平行四辺形の4辺の各々がUNLEによるラインである平行四辺形12Aにより近似するならば、容易となる。最良の平行四辺形12Aは、通常、フットプリント12aの最遠対向コーナー間のより大きな対角線に近接した主対角線14aを有する境界ボックス14におけるものである。そのとき、点(例えば、特定のテクセル中心12b')は、全てのUNLEが点12b'について正の値を生じるならば、フットプリント平行四辺形12Aの中にある。4ビットインジケータワードが、各被試験点に付してある。各ビットは、関連規則的UNLEにおける試験の結果により設定される(例えば、ワードは、ビットWXYZ(但し、Wは第一(L1)UNLE試験の符号であり、Xは同じ点についてのL2/第二UNLE試験の符号であり、YはL3UNLE試験の符号であり、ZはラインL4についての第四UNLE試験結果である)。4つのUNLE試験のうちの特定のものが正である場合には、この点に関連する4ビットのうちのひとつを1に設定し、さもなければ、負の結果では、0ビットとなる。このプロセスは、4つの線方程式の各々について行われ、(1111バイナリ又は(15デシマル)の値を有するこれらの点は、ピクセルフットプリント内にあるとされる。例えば、第一テクセル中心点12b'-1は、(---++)又はバイナリ0011=3デシマルのUNLE結果を有し、フットプリント12aの外であり、一方、別の中心点12b'-2は(++++)又はバイナリ1111のUNLE結果を有し、修正フットプリント12Aの中である。もしピクセルフットプリント12aを、同じ係数を有する2組の平行線方程式を有する、2組の間隔のあいた平行側面を有する平行四辺形12Aとすれば、4つのUNLEを2つのUNLE方程式に単純化できる。上記したように、

$$L = L_o + (L_s \times \Delta s) + (L_t \times \Delta t)$$

(但し、( $s_a$ 、 $t_a$ )での第一の点及び( $s_b$ 、 $t_b$ )での別の点では、 $\Delta s = s_b - s_a$  及び  $\Delta t = t_b - t_a$  であり；( $s_1$ 、 $t_1$ )及び( $s_2$ 、 $t_2$ )では、これは  $L_1 = ((s_2 \times t_1) - (s_1 \times t_2)) + ((s_1 - s_2) \times \Delta s) + ((t_2 - t_1) \times \Delta t)$  となる)。上記ラインL1に平行なラインL3は、( $s_1 + a$ 、 $t_1 + b$ )及び( $s_2 + a$ 、 $t_2 + b$ )(但し、 $a$ 及び $b$ は、オフセット定数である)を有する。したがって、平行線の第一項は、 $L'_o = (s_2 + a) \times (t_1 + b) - (s_1 + a) \times (t_2 + b)$  であり、これは、 $L'_o = L_o - (L_s \times a) - (L_t \times b)$  と同じであるか、又は  $L'_o = L_o + k$  である。但

し、 $k = -(L_s \times a) - (L_t \times b)$ 。しかしながら、 $L'_s = L_s$  及び  $L'_t = L_t$  であるので、 $L' = L + \Delta$  (但し、 $\Delta$ は、( $a$ 、 $b$ )オフセットである)。ステップ39において、「s」及び「t」の両方向におけるラスタ様均一インクレメンテーション下で、フットプリント12Aに包囲されたテクセル12bは、UNLEを用いて射影ピクセルフットプリントにより包囲されているテクセル中心点12b'の数を求めることにより分かる。全て1(即ち、11112)を含むバイナリワードを特定のテクセルについて計算する場合には、そのテクセル中心は包囲され、対象テクセルはピクセル属性に寄与するか、さもなければ、フットプリントの外であり、テクセルは、検数に寄与しない。CIGのグラフィックプロセッサは、各テクセル列沿って左から右(即ち、必要範囲全体を通過するs値を変化しながら)に処理してから、列数のインクレメンテーション(即ち、t値を変化)を行って、テクセルごとに新しいUNLEに基づく結果を計算する。テクスチャスペースにおける境界ボックス14の上左手コーナーにおいて開始し、4つのUNLEの各々について完全方程式を計算して、フットプリントに対する各テクセル中心点位置を決定する。右方向に次のテクセル中心点の値を、各方程式について前に求めた値をセーブし且つセーブされた値に関連 $\Delta s$ だけ増加することにより計算する。このインクレメンテーション法を、列を横断して継続し、同じサブプロセスを、関連 $\Delta t$ 量だけ増加しながら各続いての列について行う。これを、境界ボックス14内の全てのテクセルを検討するまで行う。フットプリントに包囲された中心点12b'を有するテクセルのグラフィックス属性(即ち、数)を処理して、テクスチャードピクセル値を得る。工程41では、線方程式からの1111ワード結果を達成する、即ち、オンとなる(カラー/強度寄与率は1)又はテクセルがオフとなる(カラー/強度寄与率は0(無))、ピクセル合計への全てのテクセル12bの中心包囲テクセル寄与率を合計する。ピクセル射影フットプリントの総領域内に中心を有する全てのテクセルのテクスチャリング効果は、ピクセルテクスチャリングに寄与し、これは、射影ピクセルが高アスペクト比(即ち、幅又は長さのうちのひとつのフットプリント寸法が他の寸法よりもはるかに小さい)であるか、低アスペクト比(即ち、フットプリント12aの幅と長さが実質的に等しい)であるかとは無関係であることがわかるであろう。このプロセスにより、約 $\tan^{-1}(2T)$ の角度でのポリゴン間の識別がなされる。ここで、Tは、フットプリントの最大軸について選択されたLODにおけるテクセルの最小数である。短軸方向における最小臨界寸法は、エッジから中心までの距離であり、即ち、テクセル寸法のほぼ半分であり、したがって、例えば、最初の正方形ピクセルの特定のフットプリントの長軸に沿った約5テクセルを生じるLODについては、アスペクト



比は、約  $\tan^{-1}(10)$  又は約  $84^\circ$  である。工程 43 は、選択された方法で平均等を行うことにより、工程 41 での合計を適切に処理して、ピクセルテクスチャ特性を設定する。この一連の特性は、工程 45 において、いずれか他の存在因子、例えば、異なるピクセルライティング効果、に応じて修正してよい。前記ピクセルについての異方テクスチャードピクセル値を、工程 47 において CIG フレームバッファ又は別個のメモリーに保存し、工程 49 に入って、ポリゴン N についてテクスチャリングされたままのピクセル数を求める。もし追加のピクセルがあるならば、49a から出て、工程 25 に再び入る。そこで、ピクセル数  $P_{ij}$  を増加する。現在のポリゴンについて追加のピクセルが存在しない場合には、49b から出て工程 51 に入る。工程 51 において、同様の決定を、現在のシーンにおける追加のポリゴンについて行う。もし追加のポリゴンについての処理をしなければならない場合には、51a から出て、プロセスを工程 23 に戻す。もしそうでないならば、51b から出て工程 53 に入る。これでテクスチャプロセスを終了し、プロセスを次の操作に移す。図 2e において、テクスチャリングプロセスの質を向上するために、各テクセル 12b を、複数のサブテクセル 16 にさらに分割する。好ましくは、分割を、各テクセル 12b の S 及び T 方向の各々に沿って、等しいサブ分割数  $s$  として、サブテクセル 16  $s^2$  を生成する。ここで、 $s$  は、現在検討しているテクセル 12b において 2、3、4、... である。各テクセル 12b が、それ自体の中心点 12b' を有するのと同様に、各サブテクセル 16 は、それ自体の中心点 16' を有し、ここでは、サブテクセルがフットプリント 12a の中か外かについての決定に使用する。したがって、本発明では、各テクセルを  $s^2$  サブテクセルにさらに分割するのが好ましい。ここで、好ましくは  $s=2$ 、3 又は 4 であり、 $s=2$  の場合には、テクセルを 4 サブテクセル 16a ~ 16b に分割し、一方、 $s=3$  の場合には、9 サブテクセル 16a ~ 16i とし、 $s=4$  の場合には、16 サブテクセル 16a ~ 16p とする。ここで、 $s$  は、テクセルの各側に沿ったさらなる分割のいずれかの整数でよいが、 $2 \leq s \leq 4$  の範囲の正方形サブアレイでは、処理量を最小とせずに細部を最大化される（極めて多数のサブテクセルについて過剰に長い処理時間のため）。また、下に位置するポリゴンの最大アスペクト角は、主テクスチャ面方向において 5 テクセルを生成する LOD を用いて、 $4 \times 4$  サブテクセルアレイについて、ほとんど  $89^\circ$  であることができることが分かるであろう。実質的にその面においてポリゴン 12 を見るにより形成されるアスペクト比がわずかに大きいと（ポリゴン面に対する法線に対して実質的に  $90^\circ$  に等しい角度）、ポリゴンの表示シーン画像に対する寄与が実質的にゼロとなる。これは、アスペクト比  $90^\circ$  では、ポリゴンが「エッジオ

ン」で見え、実質的に消失し、寄与が生じないので、望ましい。各テクセルを、4 つの UNL E の符号に対しての同種の比較に応答して、オンとされたサブテクセルの数により評価する。ここで、1 ピクセル当たり  $s^2$  の比較がなされ、 $s^2$  組の 4 ビット比較ワード (WXYZ) を検討する。もしいずれか一つのサブテクセルについての比較が、(1111) 以外である場合には、そのサブテクセルはピクセルフットプリント 12a には包囲されないで、検討されない。サブテクセル比較ワードが (1111) である場合のみ、サブテクセル値が包囲され、ピクセル検数にカウントされる。したがって、例えば、 $2 \times 2$  サブ分割テクセルにおいて、1 サブテクセルが「オン」の場合（即ち、包囲され且つカウントされるべき）、テクセル値は  $1/4$  であり、2 サブテクセルがオンの場合、テクセル値は  $1/2$  であり、もし 3 サブテクセルがオンであるならば、テクセル値  $3/4$  であり、そして 4 サブテクセルの全てがオンの場合には、値は、勿論 1 フルテクセル値である。この比較は、修正工程 39 においてなされる。フットプリントに包囲されたサブテクセルの数に基づいて、テクセルを、ピクセルがテクスチャリングされるテクスチャカラーに寄与する加重値に割り当てる。上記したように、LOD は、射影ピクセルフットプリント 12a の長軸に沿った所定のシーケンシャルテクセル数を提供するように調整するのが好ましい。場合によっては、最微細 LOD であっても、境界ボックス主対角線 14a の大きさが、1 テクセル長未満に減少する。この場合、必要に応じて、クランピングを行ってもよい。このクランピングにより、ピクセルフットプリントの形状のその重心が維持される。ピクセルフットプリントを、次にある大きさ（即ち、拡大）にして、フットプリント 12a がこのクランピングされた境界ボックス 14 を満たすようにする。典型的には、これは、視点 10 がテクスチャポリゴン 12a に極めて近くに移動してテクスチャが非常にブロックィになるのを防止するときに起きる。したがって、もし境界ボックスの最大サイズが最微細 LOD におけるテクセル側よりも小さいならば、フットプリントと境界ボックスの両方を、最大サイズがテクセルの大きさとなるまで拡大して、ピクセルフットプリントの最小サイズを 1 テクセルにクランピングし、さもなければ、テクセルをクローズインビューをする。図 3 において、本方法は、グラフィックプロセッサとしての役割を果たすプログラム化汎用コンピュータで実施することができるが、専用計算機 60 又は ASIC も、同等に使用して処理及び演算を実施してもよい。2 点からの各 UNL E についての係数を計算するには、2 つのデジタル乗算器と 3 つのデジタル加算器が必要である。演算の精度は、グリッドサイズの関数であり、整数演算を使用できる。スクリーンスペースピクセル位置  $P_{ij}$  を入力手段 60a に設けて、ピクセルテクスチャマッピング手段 62 の入力 62a と、ピクセル

勾配計算手段64の入力64aとの両方に通じるようにする。手段62は、視線10a（例えば、観測スペースフレームに対する角度）を特徴付ける入力60bデータから受け取る別の一連の入力62bを有することにより、一連の出力62cがテクスチャスペースフットプリント点Pstを確立するデータを提供できるようになっている。出力62cでのデータを、第一加算手段66への第一の一連の入力66aに結合する。これは、手段64の出力64bからの4つのテクセルスペース勾配を受け取る第二の一連の4つの入力66bを有する。また、手段64は、所望ならば、入力64cで入力60bデータを受け取ってもよい。4つの加算手段の出力66cは、それぞれ、4つのピクセルフットプリント点p1～p4の（s、t）座標を有する。入力68aに役立つこれらのフットプリント点から、最大サイズサブプロセッサ手段68は、Smax及びTmax等を計算して、境界ボックス14及びその主方向寸法14aを決定する。これらのデータは、出力68bを通過して除算手段70の第一入力70aに至る。入力70bで提供されたLODグリッド情報を、入力70aでの主寸法値の除数として使用することにより、射影ピクセルフットプリントの主方向に沿ってカバーされたテクセル数を、出力70cで得られるようにする。この数を、log2ルックアップテーブル手段72の入力72aでアドレスとして使用する。log2ルックアップテーブル手段72の入力72aは、グラフィックスアクセラレータテクスチャメモリー手段73のアドレス入力73aへの出力72bで使用されるべきLODを選択する。手段73は、バス73bで、テクセル/サブテクセルコンテンツ/属性データを出力する。適切なLODを決定して、出力66cで境界ボックスコーナー点情報を、UNLEの部分的な項を提供するように構成された一連の加算器74及び乗算器76に加える。これらの出力を、別の加算手段78に供給して4つのUNLEを計算する。これらは、コンパレータ手段80の第一の一連の入力80aに加えられ、別の一連のコンパレータ手段入力80bに現れる、テクスチャメモリー手段73の出力73bからのテクセル/サブテクセルデータと比較する。もしテクセル12bをQサブテクセル16に分割した場合には、Qの異なるサブテクセル（ここでは、Q=4サブテクセルの2x2アレイ）の各々を、UNLEと、一連の比較手段（ここでは、4つの手段80-1～80-4）のうちの異なる一つにより比較する。もしサブテクセル16x（但し、X=a、b、c又はdでUNLEワードが15Hである）である場合には、そのサブテクセルについてのカラー属性データを、サブテクセルアキュムレータ手段82に入れる。全てのQサブテクセルを検討した後、アキュムレータ手段の出力82aは、次に処理される一つのテクセルについてのデータを有する。このデータは、「サムオールテクセル」アキュムレータ手段84に

通して、現在射影されているピクセルについての総テクセル寄与データを出力60bに加える。出力60bでのピクセルデータを、さらに処理するか、フレームバッファ等の記憶装置に直接入れて、CIGディスプレイ手段に対してラスタ（又は他の）走査し、観測者10により観察されるようにする。本発明のシルエット解析による異方性テクスチャマッピング用の新規な方法及び装置の好ましい実施態様を詳細に説明したが、数多くの変更及び修正が可能なのは、当業者には明らかであろう。したがって、本発明は、請求の範囲の範囲によってのみ限定され、本明細書に記載の詳細な説明及び手段によっては限定されない。

#### 【図面の簡単な説明】

【図1】A. 画像スペースにおける表示スクリーン面を介してテクスチャードポリゴンを見る観測者錐台であって、本発明の方法の背景を理解するのに有用な観測者錐台の等角図である。

B. 正方形表示スペースピクセルのフットプリントが射影によるテクスチャスペース一部分を示す平面図である。

【図2】本発明の方法の好ましい実施態様を示す流れ図である。

A. 表示スペーススクリーン及びその上のピクセルシルエット並びにテクスチャスペースアレイの一部分及びその上の射影ピクセルフットプリントの並行比較を示す図である。

B. 射影ピクセルとその空間デリバティブの一部分を有し、本発明の方法を使用するのに必要ないくつかの概念を理解するのに有用である、テクスチャスペースを示す図である。

C. 射影ピクセルを有するテクスチャスペースと本発明の方法により加えられた境界ボックスを示す別の図である。

D. 本発明の方法で使用される非標準化線方程式を生成する4つの境界線を示す、テクスチャスペースにおけるピクセルフットプリントを示す図である。

E. テクセルごとに4、9又は16サブテクセルを使用してテクスチャの解像度を向上させることを示した、一連のサブ分割テクセルの説明図である。

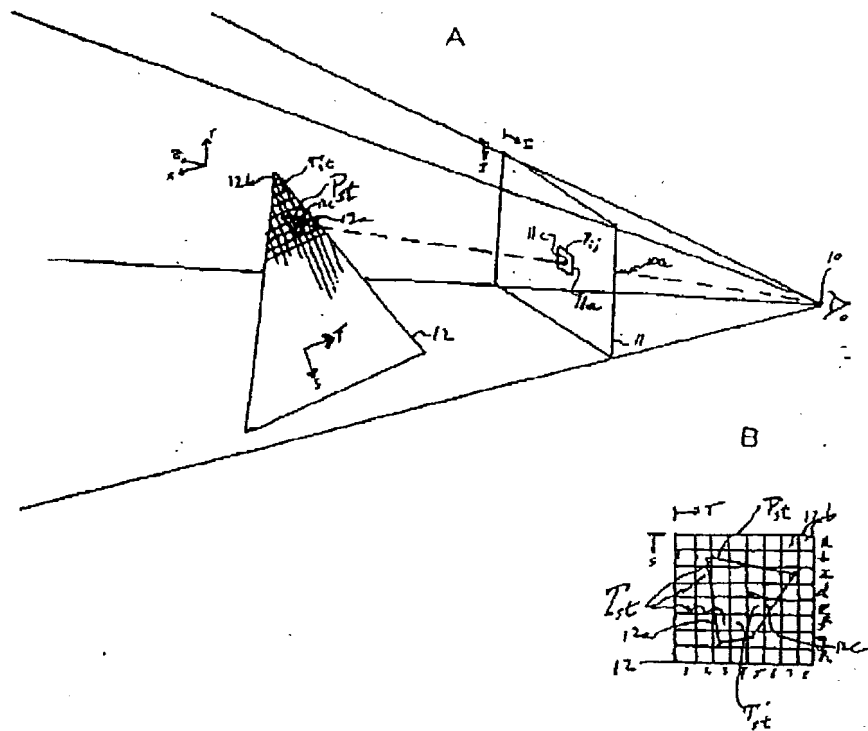
【図3】好ましい方法を実施するのに使用できる一つのハードウェアを示した概略ブロック図である。

#### 【符号の説明】

- 10 視点
- 10a 視線
- 11 スクリーン表示面
- 11a ピクセル
- 12 テクセル面
- 12a テクスチャスペースピクセル射影フットプリント
- 12b テクセル

- |                      |                       |
|----------------------|-----------------------|
| 14 境界BBボックス          | 70 除算手段               |
| 16 サブテクセル            | 72 log2ルックアップテーブル手段   |
| 60 専用計算機             | 73 テクスチャメモリー手段        |
| 60a 入力手段             | 73b バス                |
| 62 ピクセル-テクスチャマッピング手段 | 80 コンパレータ手段           |
| 62c 出力               | 84 サムオールテクセルアキュムレータ手段 |
| 64 ピクセル勾配計算手段        | ○ 観測者                 |
| 66 第一加算手段            | Pij アレイピクセル           |
| 68 最大サイズサブプロセッサ手段    | TST アレイテクセル           |

【図1】





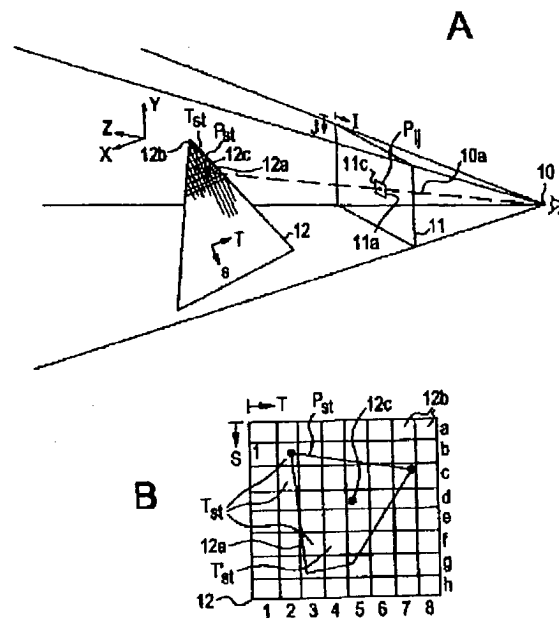
1082レックアーアップデータ手段

68a 68b 68c 68d 68e 68f 68g 68h 68i 68j 68k 68l 68m 68n 68o 68p 68q 68r 68s 68t 68u 68v 68w 68x 68y 68z 68aa 68ab 68ac 68ad 68ae 68af 68ag 68ah 68ai 68aj 68ak 68al 68am 68an 68ao 68ap 68aq 68ar 68as 68at 68au 68av 68aw 68ax 68ay 68az 68ba 68bb 68bc 68bd 68be 68bf 68bg 68bh 68bi 68bj 68bk 68bl 68bm 68bn 68bo 68bp 68bq 68br 68bs 68bt 68bu 68bv 68bw 68bx 68by 68bz 68ca 68cb 68cc 68cd 68ce 68cf 68cg 68ch 68ci 68cj 68ck 68cl 68cm 68cn 68co 68cp 68cq 68cr 68cs 68ct 68cu 68cv 68cw 68cx 68cy 68cz 68da 68db 68dc 68dd 68de 68df 68dg 68dh 68di 68dj 68dk 68dl 68dm 68dn 68do 68dp 68dq 68dr 68ds 68dt 68du 68dv 68dw 68dx 68dy 68dz 68ea 68eb 68ec 68ed 68ee 68ef 68eg 68eh 68ei 68ej 68ek 68el 68em 68en 68eo 68ep 68eq 68er 68es 68et 68eu 68ev 68ew 68ex 68ey 68ez 68fa 68fb 68fc 68fd 68fe 68ff 68fg 68fh 68fi 68fj 68fk 68fl 68fm 68fn 68fo 68fp 68fq 68fr 68fs 68ft 68fu 68fv 68fw 68fx 68fy 68fz 68ga 68gb 68gc 68gd 68ge 68gf 68gg 68gh 68gi 68gj 68gk 68gl 68gm 68gn 68go 68gp 68gq 68gr 68gs 68gt 68gu 68gv 68gw 68gx 68gy 68gz 68ha 68hb 68hc 68hd 68he 68hf 68hg 68hh 68hi 68hj 68hk 68hl 68hm 68hn 68ho 68hp 68hq 68hr 68hs 68ht 68hu 68hv 68hw 68hx 68hy 68hz 68ia 68ib 68ic 68id 68ie 68if 68ig 68ih 68ii 68ij 68ik 68il 68im 68in 68io 68ip 68iq 68ir 68is 68it 68iu 68iv 68iw 68ix 68iy 68iz 68ja 68jb 68jc 68jd 68je 68jf 68jg 68jh 68ji 68jj 68jk 68jl 68jm 68jn 68jo 68jp 68jq 68jr 68js 68jt 68ju 68jv 68jw 68jx 68jy 68jz 68ka 68kb 68kc 68kd 68ke 68kf 68kg 68kh 68ki 68kj 68kl 68km 68kn 68ko 68kp 68kq 68kr 68ks 68kt 68ku 68kv 68kw 68kx 68ky 68kz 68la 68lb 68lc 68ld 68le 68lf 68lg 68lh 68li 68lj 68lk 68ll 68lm 68ln 68lo 68lp 68lq 68lr 68ls 68lt 68lu 68lv 68lw 68lx 68ly 68lz 68ma 68mb 68mc 68md 68me 68mf 68mg 68mh 68mi 68mj 68mk 68ml 68mm 68mn 68mo 68mp 68mq 68mr 68ms 68mt 68mu 68mv 68mw 68mx 68my 68mz 68na 68nb 68nc 68nd 68ne 68nf 68ng 68nh 68ni 68nj 68nk 68nl 68nm 68nn 68no 68np 68nq 68nr 68ns 68nt 68nu 68nv 68nw 68nx 68ny 68nz 68oa 68ob 68oc 68od 68oe 68of 68og 68oh 68oi 68oj 68ok 68ol 68om 68on 68oo 68op 68oq 68or 68os 68ot 68ou 68ov 68ow 68ox 68oy 68oz 68pa 68pb 68pc 68pd 68pe 68pf 68pg 68ph 68pi 68pj 68pk 68pl 68pm 68pn 68po 68pp 68pq 68pr 68ps 68pt 68pu 68pv 68pw 68px 68py 68pz 68qa 68qb 68qc 68qd 68qe 68qf 68qg 68qh 68qi 68qj 68qk 68ql 68qm 68qn 68qo 68qp 68qq 68qr 68qs 68qt 68qu 68qv 68qw 68qx 68qy 68qz 68ra 68rb 68rc 68rd 68re 68rf 68rg 68rh 68ri 68rj 68rk 68rl 68rm 68rn 68ro 68rp 68rq 68rr 68rs 68rt 68ru 68rv 68rw 68rx 68ry 68rz 68sa 68sb 68sc 68sd 68se 68sf 68sg 68sh 68si 68sj 68sk 68sl 68sm 68sn 68so 68sp 68sq 68sr 68ss 68st 68su 68sv 68sw 68sx 68sy 68sz 68ta 68tb 68tc 68td 68te 68tf 68tg 68th 68ti 68tj 68tk 68tl 68tm 68tn 68to 68tp 68tq 68tr 68ts 68tt 68tu 68tv 68tw 68tx 68ty 68tz 68ua 68ub 68uc 68ud 68ue 68uf 68ug 68uh 68ui 68uj 68uk 68ul 68um 68un 68uo 68up 68uq 68ur 68us 68ut 68uu 68uv 68uw 68ux 68uy 68uz 68va 68vb 68vc 68vd 68ve 68vf 68vg 68vh 68vi 68vj 68vk 68vl 68vm 68vn 68vo 68vp 68vq 68vr 68vs 68vt 68vu 68vv 68vw 68vx 68vy 68vz 68wa 68wb 68wc 68wd 68we 68wf 68wg 68wh 68wi 68wj 68wk 68wl 68wm 68wn 68wo 68wp 68wq 68wr 68ws 68wt 68wu 68wv 68ww 68wx 68wy 68wz 68xa 68xb 68xc 68xd 68xe 68xf 68xg 68xh 68xi 68xj 68xk 68xl 68xm 68xn 68xo 68xp 68xq 68xr 68xs 68xt 68xu 68xv 68xw 68xx 68xy 68xz 68ya 68yb 68yc 68yd 68ye 68yf 68yg 68yh 68yi 68yj 68yk 68yl 68ym 68yn 68yo 68yp 68yq 68yr 68ys 68yt 68yu 68yv 68yw 68yx 68yy 68yz 68za 68zb 68zc 68zd 68ze 68zf 68zg 68zh 68zi 68zj 68zk 68zl 68zm 68zn 68zo 68zp 68zq 68zr 68zs 68zt 68zu 68zv 68zw 68zx 68zy 68zz

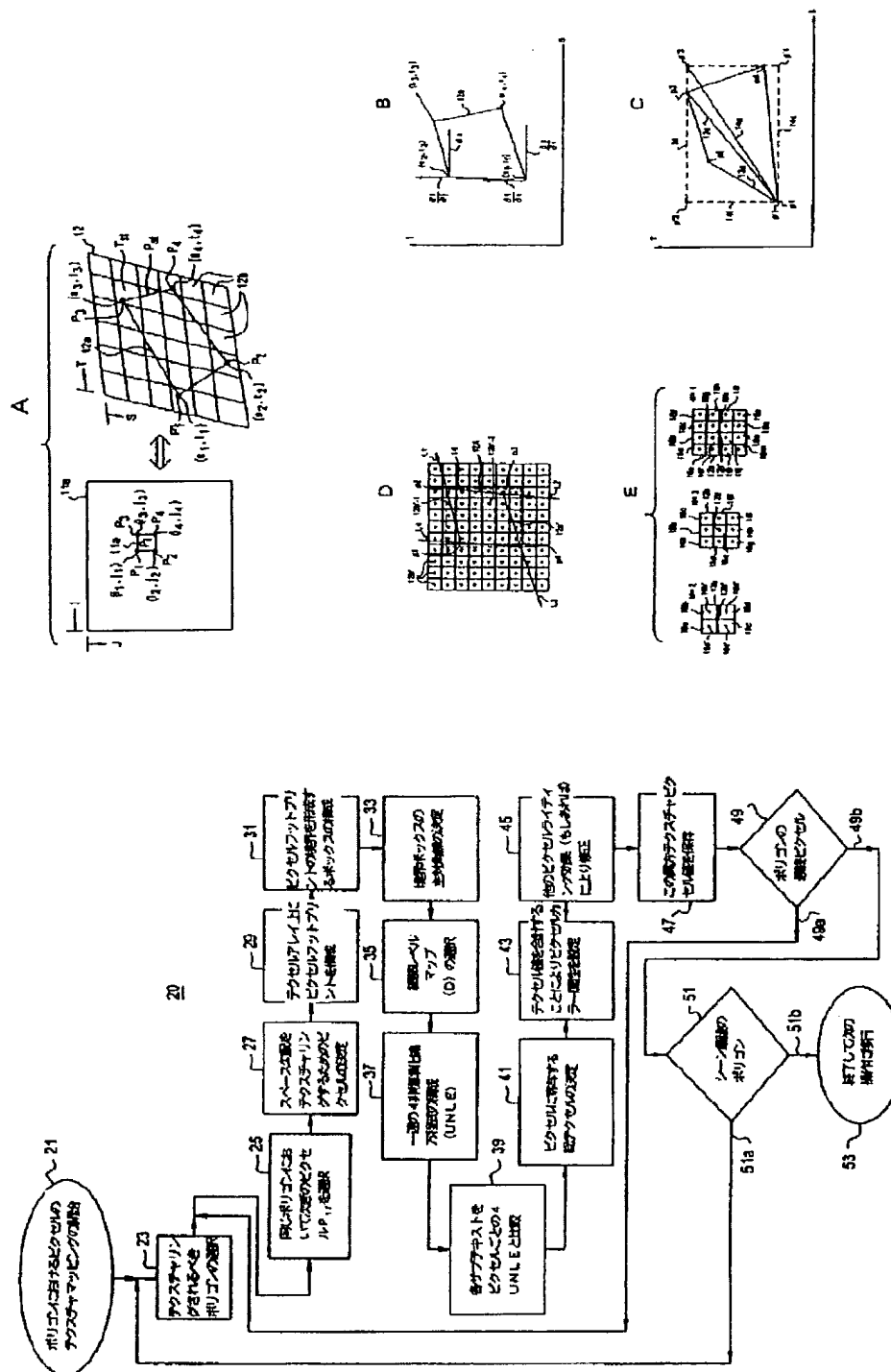
68a 68b 68c 68d 68e 68f 68g 68h 68i 68j 68k 68l 68m 68n 68o 68p 68q 68r 68s 68t 68u 68v 68w 68x 68y 68z 68aa 68ab 68ac 68ad 68ae 68af 68ag 68ah 68ai 68aj 68ak 68al 68am 68an 68ao 68ap 68aq 68ar 68as 68at 68au 68av 68aw 68ax 68ay 68az 68ba 68bb 68bc 68bd 68be 68bf 68bg 68bh 68bi 68bj 68bk 68bl 68bm 68bn 68bo 68bp 68bq 68br 68bs 68bt 68bu 68bv 68bw 68bx 68by 68bz 68ca 68cb 68cc 68cd 68ce 68cf 68cg 68ch 68ci 68cj 68ck 68cl 68cm 68cn 68co 68cp 68cq 68cr 68cs 68ct 68cu 68cv 68cw 68cx 68cy 68cz 68da 68db 68dc 68dd 68de 68df 68dg 68dh 68di 68dj 68dk 68dl 68dm 68dn 68do 68dp 68dq 68dr 68ds 68dt 68du 68dv 68dw 68dx 68dy 68dz 68ea 68eb 68ec 68ed 68ee 68ef 68eg 68eh 68ei 68ej 68ek 68el 68em 68en 68eo 68ep 68eq 68er 68es 68et 68eu 68ev 68ew 68ex 68ey 68ez 68fa 68fb 68fc 68fd 68fe 68ff 68fg 68fh 68fi 68fj 68fk 68fl 68fm 68fn 68fo 68fp 68fq 68fr 68fs 68ft 68fu 68fv 68fw 68fx 68fy 68fz 68ga 68gb 68gc 68gd 68ge 68gf 68gg 68gh 68gi 68gj 68gk 68gl 68gm 68gn 68go 68gp 68gq 68gr 68gs 68gt 68gu 68gv 68gw 68gx 68gy 68gz 68ha 68hb 68hc 68hd 68he 68hf 68hg 68hh 68hi 68hj 68hk 68hl 68hm 68hn 68ho 68hp 68hq 68hr 68hs 68ht 68hu 68hv 68hw 68hx 68hy 68hz 68ia 68ib 68ic 68id 68ie 68if 68ig 68ih 68ii 68ij 68ik 68il 68im 68in 68io 68ip 68iq 68ir 68is 68it 68iu 68iv 68iw 68ix 68iy 68iz 68ja 68jb 68jc 68jd 68je 68jf 68jg 68jh 68ji 68jj 68jk 68jl 68jm 68jn 68jo 68jp 68jq 68jr 68js 68jt 68ju 68jv 68jw 68jx 68jy 68jz 68ka 68kb 68kc 68kd 68ke 68kf 68kg 68kh 68ki 68kj 68kl 68km 68kn 68ko 68kp 68kq 68kr 68

【補正内容】

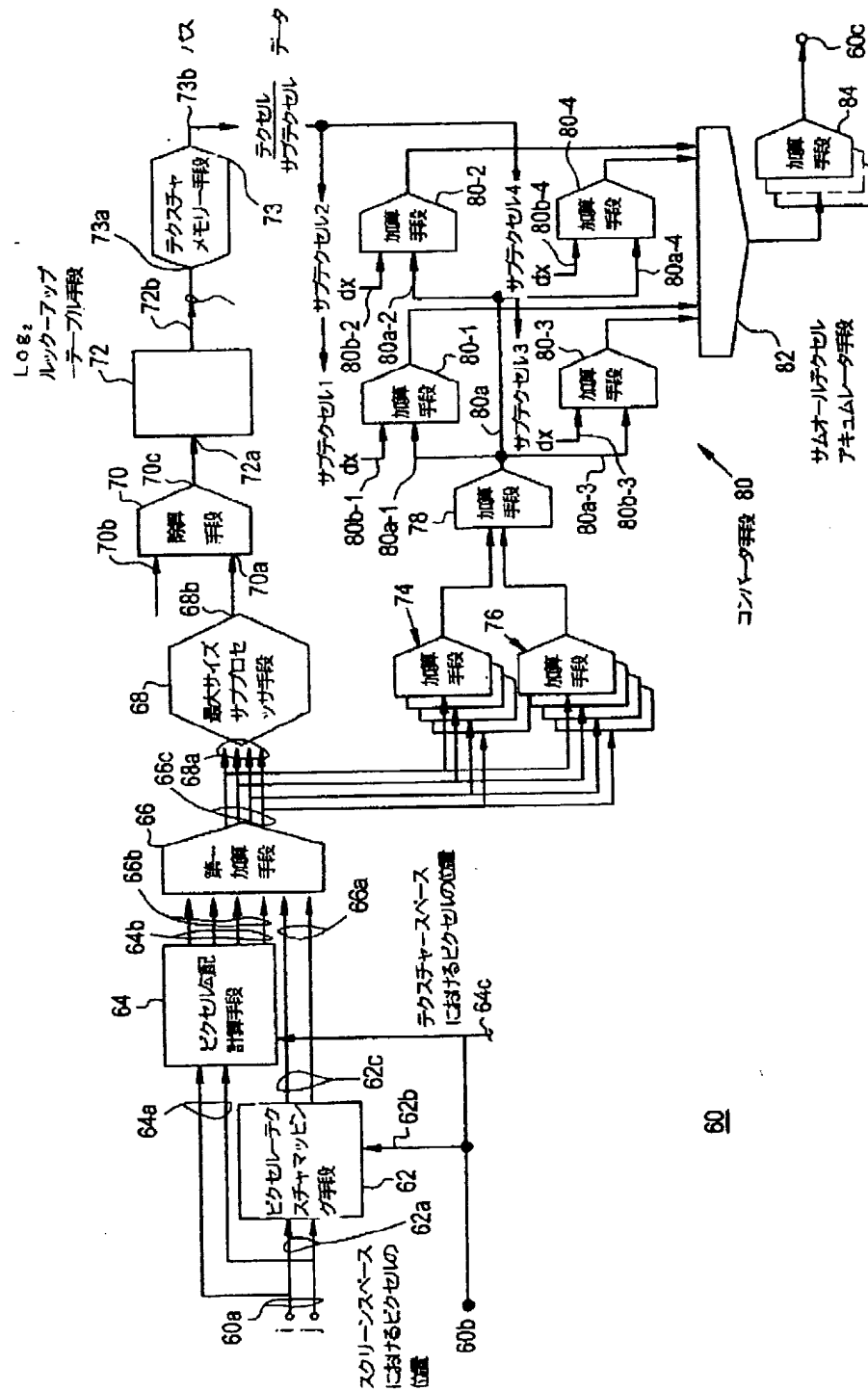
【図 1】



【図2】



【図3】





【外国語明細書】

Docket 35-GS-2781

ANISOTROPIC TEXTURE MAPPING  
USING SILHOUETTE/FOOTPRINT ANALYSIS  
IN A COMPUTER IMAGE GENERATION SYSTEM

5

The present application is directed to a computer image generation (CIG) system, also referred to as a graphics accelerator, for the display of two-dimensional (2D) representations of three-dimensional (3D) scenes and, more particularly, to novel methods and apparatus for mapping texture anisotropically to each display pixel by silhouette analysis of each display space pixel required for each of a plurality of polygons forming each, preferably real-time, scene on a CIG display, in a computational efficient manner.

20 Background of the Invention

It is now well known to provide a computer image generation system, and associated processes, for generation of 3D image scenes upon a 2D display; each scene may contain many thousands (typically, over one million) of individual picture elements, called pixels, with each pixel having visual attributes unique to that corresponding small visible area in a 2D image space representation of the 3D object space then observable by a viewer situated with a certain orientation relative to the objects.

*Docket 35-GS-2782*

The objects themselves are viewed from a known observation point, and may thus be represented by a set of planar surface polygons, which may be textured, by projection into texture space  
5 for convolution with the texture elements (texels) of selected texture patterns, prior to display. The various aspects of a CIG textured-polygon process may be had by reference to available texts and to previous Letters Patent,  
10 including U.S. patent 4,727,365 (Advanced Video Object Generator, issued Feb. 28, 1988) and U.S. patent 4,974,176 (Microtexture for Close-In Detail, issued Nov. 27, 1990), both of which patents are assigned to the assignee of the  
15 present application and are incorporated herein in their entireties by reference, along with all of the references cited therein.

It has been observed that most texture mapping approaches used in real time and with  
20 interactive graphic systems can not achieve a quality texture mapping for shallow viewing angles of textured objects (relative to the plane of the object polygon), i.e. where the object to be textured has a high aspect ratio  
25 (i.e., with a high angle of viewing incidence, generally greater than 45°, with respect to the polygon normal). An example of this is the "White Line Down the Center of the Road" problem: in the real world, the broken white  
30 road line is a series of white line segments each about 10 feet long and 5 inches wide with

Docket 35-GS-2782

each white line segment separated by 10 feet of black pavement. Display of a 2D representation of a real 3D white road line would try to achieve two conflicting goals: first, to  
5 prevent the white line from disappearing as an observer looks down the road; and second, to prevent the resulting displayed white line from scintillating - here, scintillation is manifest as the appearance and disappearance of  
10 different sets (say, 30 feet lengths) of several white-line segments along the road. Another related example of these problems is the change in building features that are sharp when facing the viewer and either scintillate  
15 or take on a soft appearance when viewed at a high aspect ratio (shallow angle to the viewer). Most of these problems are caused by a failure to account for the display pixel (which usually has a rectangular, and often  
20 square, silhouette in image space) having an elongated footprint when mapped into texture space.

Texture maps are often organized into Level of Detail (LOD) sets. Typically, each  
25 coarser-level member of the set is a texture map that is a 2:1 reduction of the nextmost-finer set member above it; thus, a level D texture map will have 4 (or 2x2) texture elements (texels) for each texel in the texture  
30 map for the next-coarser level D+1. So, if the finest, first LOD (D=0) map has an array of

*Docket 35-GS-2782*

512-by-512 texels, the next-coarser LOD ( $D=1$ ) is a texture map that is made up of an array of 256-by-256 (or  $256 \times 256$ ) texels. This process is carried out, for this example, until LOD 9  
5 yields a map with a single  $1 \times 1$  texel.

In hitherto normal texture-to-pixel mapping, the effective LOD of each pixel is calculated for both of the width and length of that pixel as projected in texture space and  
10 the larger of the two effective LOD numbers is selected as that pixel's effective LOD. Next, two texture maps with LODs that bracket the effective LOD of the involved pixel are selected and an interpolation between two  
15 adjacent LOD texture maps is performed, using one of 3 interpolation strategies, is performed to calculate the value of the mapped pixel. The three interpolation methods are: nearest neighbor interpolation, bi-linear  
20 interpolation, or tri-linear interpolation.

Because the larger of width or length of the modified pixel, as projected in texture space, is used to determine the pixel's effective texture space LOD, the projected  
25 pixel effective size is forced to have the same texture space size in both its length and width directions. These texture methods are said to be isotropic because the LOD of the mapped pixel are always the same in both directions in  
30 texture space. However, when normally-rectangular (preferably, square) image space

*Docket 35-GS-2782*

pixels are projected onto a object-space polygon which is skewed to some degree with respect to the view axis (itself normal to the image/display space plane), each projected or  
5 (texture-) mapped pixel becomes elongated because of the shallow projection angle; if such elongation (the ratio of the greater projected dimension to the lesser projected dimension) is beyond some selected limit (say,  
10 3:1) the well known isotropic procedures, using the fiction that length equals width, can not be accommodated. This is the anisotropic case, where the projected pixel footprint dimensions, and therefore the associated LODs, can not be  
15 the same in both directions without causing undesired visual artifacts.

It is highly desirable to provide anisotropic texture methods that account for a projected pixel elongated shape in texture  
20 space. Several such methods (usually for non-real-time systems) have been implemented by approaches which fall into one of two broad categories: 1) Methods for convoluting the pixel texture-space-projection with the texture  
25 values, for producing very accurate, but computationally very expensive, results; and 2) Methods using the storage of pre-processed and pre-filtered textured maps, which require less computations but are also less flexible  
30 and precise, and more memory intensive.

*Docket 35-GS-2782*

Because of the increasing demand for more and more different textures maps to be available for real-time and interactive systems, the less memory intensive approaches are more desirable. In the past, other schemes for rendering anisotropic texture have been suggested (such as a procedure, used almost a decade ago by our predecessor simulation group, of sampling the texture along only the single greater diagonal of the anisotropic pixel) have hitherto been found to either provide insufficient resolution of the anisotropic texturing problems or be too costly (in terms of either necessary hardware or computational time).

Accordingly, provision of novel methods for mapping texture anisotropically to each display pixel required for each of a plurality of polygons forming each, preferably real-time, scene on a CIG display in a computational efficient manner, is highly desirable.

#### Brief Summary of the Invention

In accordance with the invention, a method for anisotropically mapping texture information to each pixel required for display of each of plural polygons forming each scene on a CIG display in a computational efficient manner, uses the footprint of each transformed pixel, as mapped into the texture array, to determine an anisotropic texture space area having both a

*Docket 35-GS-2782*

related length and a related width, in which complete texels and texels fractions are covered by that pixel footprint. Use of a set of un-normalized line equations characterizing a particular pixel footprint is used to determine whether each texel in the vicinity of that pixel footprint is either enclosed or avoided by (i.e. inside or outside of) that projected pixel footprint. By use of an isotropic 2D area, rather than a 1D line (which is only proper for very high aspect ratios {typically greater than  $70^\circ$ }), this method allows for tallying of the included full texels or texel fractions, so as to determine the color and/or translucency of the pixel. Since all values are determined incrementally, my methods initially require the computation of a few values, followed by a simple incrementing procedure to tally which texels are included in the footprint and which are not. If the texels are subdivided into 2 or more subtexels, each subtexel can also be determined to be inside or outside of the pixel footprint, permitting more accurate texture mapping. A pixel is mapped into a higher resolution texture map than its effective isotropic LOD would indicate. The projected pixel overlaps a number of texels in all direction.

Accordingly, it is one object of the present invention to provide novel method and apparatus for the anisotropic texturing of

Docket 35-GS-2782

image space pixels using an areal projection and silhouette analysis.

5 This and other objects of the present invention will become apparent to those skilled in the art, upon consideration of the appended drawings and reading of the associated detailed description of a presently preferred embodiment.

10 Brief Description of the Drawings

Figure 1a is an isometric view of an observer frustum viewing a textured polygon through the display screen plane in image space, and useful in appreciating the background of the methods of the present invention;

Figure 1b is a plan view of a portion of texture space upon which the footprint of a square display-space pixel has been projected;

20 Figure 2 is a flow diagram for one presently preferred embodiment of the methods of the present invention;

Figure 2a is a side-by-side comparison of the display space screen and a pixel silhouette thereon, and a portion of the texture space array and the projected pixel footprint thereon;

25 Figure 2b is a view of texture space, with the projected pixel and some of its spatial derivatives, and useful in appreciation of



Docket 35-GS-2782

several concepts needed for use of the methods of the present invention;

5 Figure 2c is another view of texture space with the projected pixel, and of the bounding box added by the methods of the present invention;

10 Figure 2d is a view of a pixel footprint in texture space, showing the four bounding lines from which un-normalized line equations will be generated for use in the methods of the present invention;

15 Figure 2e is a set of subdivided texels, illustrating the use of 4, 9 or 16 subtexels for each texel to be processed, to generate better texture resolution; and

Figure 3 is a schematic block diagram of one possible hardware implementation of the preferred method.

20 Detailed Description of a Presently Proffered Embodiment

Referring initially to Figure 1a, an observer O is at viewpoint 10, viewing a display plane 11 which is the image, or  
25 display, space characterized by axes I and J; this plane is made up of an array of display picture elements, or pixels, 11a of which a general array pixel  $P_{IJ}$  is denominated by its respective I and J axial dimensions. A viewline  
30 10a passes through the center 11c of the pixel 11a silhouette and is projected as the center

*Docket 35-GS-2782*

12c of a footprint 12a of the pixel 11a as projected onto a plane 12 in texture space. Plane 12 will, in general, be an object-space polygon as projected into texture space and will have an array of texture elements, or texels, 12b thereon; the textured polygon plane 12 has axes S and T, so that a general array texel  $T_{st}$  is denominated by its S and T axial dimensions. It will be seen that the generally rectangular, often square, silhouette of pixel 11a will, because the underlying polygon 12 is generally not parallel to the screen plane 11, be projected upon plane 12 as a quadrilateral pixel  $P_{st}$  footprint 12a typically having four unequal sides as well as unequal corners which typically are not right angles.

The texturing problem can be understood by reference to Figure 1b, wherein the texel plane 12 is now orthogonal to the reader's sightline, so that the texels 12b of the array are rectangular, and preferably square, but the projected pixel  $P_{st}$  footprint 12a has a distorted quadrilateral shape. I have selected the texture level-of-detail (LOD) such that the size of each texel is generally less than the size of the projected pixel  $P_{st}$ ; thus, pixel 12a partially covers at least one texel  $T_{st}$  and may completely cover one or more other texels  $T'_{st}$ . Here, the texel rows, in the S direction, have been labeled a through h, while the texel columns, in the T direction, have been labeled

*Docket 35-GS-2782*

1 through 8: the footprint 12a fully covers  
texels 3c, 4c, 3d, 4d, 5d, 4e, 5e and 4f, but  
only (possibly) portions of texels 2b, 3b, 4b,  
5b, 2c, 7c, 3e, 6e, 3f, 5f, 6f, 3g, 4g and 5g,  
5 each of which texels may have a unique set of  
visual attributes, e.g., color/transparency  
values. The problem, as will be recognized by  
those skilled in this art, is to properly  
account for the contributions of all texels  
10 either partially-enclosed or fully-enclosed by  
the total area of particular pixel  $P_{ix}$  footprint  
12a, in order to determine the proper color,  
translucency and the like visible attributes  
for display at the display space site of that  
15 pixel  $P_{ij}$ .

Referring now to Figure 2, in one  
presently preferred embodiment of my novel  
process, a 3D computer image generator (CIG)  
may have a graphics processor (which may be a  
20 dedicated hardwired logic engine or a software-  
programmed general computer) which carries out  
the isotropic-texturing silhouette/footprint  
analysis procedure shown in flow diagram 20.  
The process starts at step 21. Step 23 selects  
25 the next polygon 12 to be textured; in this  
embodiment, pixels may be grouped for each  
particular polygon to be rendered by the  
graphics engine, although other methods of  
pixel operation can equally as well be  
30 utilized. In step 25, procedure 20 selects the  
pair of I and J values of the next pixel  $P_{ij}$ .

Docket 35-GS-2782

which is to be textured, in the presently-  
selected polygon N. This selection effectively  
determines both the image-space coordinates  
(i,j) of the pixel  $P_i$ , corners and the corner  
5 coordinates (s,t) of that pixel  $P_{st}$  as mapped  
into texture space. Referring also to Figure  
2a, a selected four-cornered quadrilateral  
pixel 11a has given corners  $p_1$  through  $p_4$ , each  
with its associated I,J coordinates, i.e.  $p_1$  is  
10 at (i1,j1),  $p_2$  is at (i2,j2),  $p_3$  is at (i3,j3)  
and  $p_4$  is at (i4,j4). The display pixel  $P_i$ ,  
silhouette 11a is transformed to the texel  
space pixel  $P_{st}$  footprint 12a, and each of the  
image space corners  $p_i$  is respectively  
15 transformed into an associated texture space  
corner  $p_{st}$ , i.e.  $p_1$  is at (s1,t1),  $p_2$  is at  
(s2,t2),  $p_3$  is at (s3,t3) and  $p_4$  is at (s4,t4).  
In step 27, the pixel-to-texel spatial  
gradients are determined and the pixel  
20 footprint is constructed on the texel array in  
step 29. As shown in Figure 2b, the mapping  
gradients, from screen space to texture space,  
are found; the partial spatial derivatives  $\partial s/\partial i$   
and  $\partial t/\partial i$  are found with respect to first corner  
25 (s1, t1) and partial spatial derivatives  $\partial s/\partial j$   
and  $\partial t/\partial j$  are then found with respect to second  
corner (s2, t2). Given these partial  
derivatives, the (i,j) coordinates of each  
pixel  $p_{(i,j)}$  can be mapped (which mapping is  
30 indicated by the  $\rightarrow$  symbol) into texture space

Docket 35-GS-2782

as  $p_{(i,t)}$ , i.e.  $p(i1,j1) \rightarrow p(s1,t1)$ ;  $p(i2,j2) \rightarrow p(s2,t2)$ ;  $p(i3,j3) \rightarrow p(s3,t3)$ ; and  $p(i4,j4) \rightarrow p(s4,t4)$ . If the first corner  $(s1,t1)$  is found by the somewhat lengthy projection-

- 5 transformation process, the other three corners can be found from a knowledge of the sidelengths in image space and the mapping gradients; for the usual situation, where each pixel has unit sidelengths acting as  
 10 multipliers of the associated gradient terms, these other corner locations are:

$$(s2,t2) = (s1 + \partial s / \partial i, t1 + \partial t / \partial i),$$

$$(s3,t3) = (s2 + \partial s / \partial j, t2 + \partial t / \partial j), \text{ and}$$

$$(s4,t4) = (s1 + \partial s / \partial j, t1 + \partial t / \partial j).$$

- 15 The complete texture space pixel projected footprint 12a can be constructed by the ordered interconnection of these four corners.

- Referring now also to Figure 2c, step 31 is entered to compute a bounding BB box 14  
 20 which is a rectangular box just touching one of the corner points  $p$  at the greatest and least  $S$  values and at the greatest and least  $T$  values of the footprint, and thus encloses the pixel footprint 12a. Then, in step 33, there is  
 25 calculated the bounding box main diagonal 14a, between the box 14 diagonally-opposed corner pair (here, corners  $p'1$  and  $p'3$ ) associated with, and closest to, the furthest-apart opposite corners (here, corners  $p1$  and  $p3$ ,  
 30 which are further apart than opposite corners

Docket 35-GS-2782

p2 and p4) forming a main diagonal 12d of the original footprint 12a; it will be seen that diagonal 14a is generally congruous to, but not identical with, the footprint main diagonal 12d. Since the bounding box 14 is determined by finding the maximum and minimum extents of the footprint in each of the S and T directions, the process of step 31 can specifically be:

$s_{max} = \max(s1, s2, s3, s4)$  and  
 $s_{min} = \min(s1, s2, s3, s4)$   
 while  $t_{max} = \max(t1, t2, t3, t4)$  and  
 $t_{min} = \min(t1, t2, t3, t4)$

where max or min signify respectively the greatest or least one of the quantities within the following parentheses. Here, it is seen that  $s_{max}=s4$  and  $s_{min}=s1$ , while  $t_{max}=t3$  and  $t_{min}=t1$ . From the foregoing, the graphics processor can next calculate the box S range  $R_s = (s_{max} - s_{min})$  or  $(s4 - s1)$  and the box T range  $R_t = (t_{max} - t_{min})$  or  $(t3 - t1)$ .

The major direction of the bounding box is determined, in step 33, by comparison of the maxima of the S extent and the T extent; the larger of range  $R_s$  and range  $R_t$  set the major direction  $D_M$ ; or,  $D_{max} = \max(R_s, R_t)$  and direction,  $D_M = R_s$  if  $D_{max} = R_s$  (i.e., if  $R_s > R_t$ ), or  $D_M = R_t$  if  $D_{max} = R_t$  (i.e., if  $R_t > R_s$ ).

Next, step 35 is entered and the proper LOD "D" is determined. To simplify the calculations, I have found that the quadrilateral footprint 12a can be approximated

*Docket 35-GS-2782*

by a properly-sized parallelogram 12A with a minimum of visible error, if a plurality of texels 12b cover each projected pixel footprint 12a. I therefore want the pixel footprint bounding box 14 area to cover several texels in at least one box direction (Figure 2d); it is preferable to cover a plurality in both axial directions, although projections with high aspect ratios may preclude this and allow plural texel coverage only along one axial direction. Note that my areal anisotropic methods want to have at least one texel, and preferably a greater number of texels, even in the "minor" direction, so as to truly have an area of coverage, rather than sampling coverage only along a line; if high aspect ratio footprints are routinely operated upon, use of the subtexel situation, discussed hereinbelow, is usually warranted, as this allows a second dimension to be used with an extent measured as a fraction of a texel, and allows a box 14 with a vanishing small minor dimension (less than one subtexel, as might be found by a polygon plane seen edge-on, at about 90°) to vanish.

The number  $T_s$  of texels 12b spanned by the S-axis side 14s of the box is calculated and compared to the calculated number  $T_t$  of texels spanned by the T-axis side of box 14. The LOD number D can be varied to have a desired number in the vicinity of either  $T_s$  or  $T_t$ ; once a good value of D is found, the proper texture grid

*Docket 35-GS-2782*

for that LOD is extracted from memory for use  
in the remaining steps. The texture grid  
placement depends on whether the grid LOD is  
odd or even, which is a measure of the maximum  
5 number of texels spanning the pixel, as the  
finest resolution LOD  $D$  is set by examination  
of the number  $T_p$  of the covered texels in one  
direction of the texel grid, as long as the  
condition  $(T_p/2^D) \leq D_{max}$  is met. Typically, I try  
10 to have 3-5 texels span the pixel along the  
longer of its  $S$  and  $T$  ranges.

In step 37 (referring again to Figure 2d),  
my methods determine whether a texel 12b is  
either inside the footprint (and therefore has  
15 its color and translucency attributes counted  
in the pixel 12a) or is outside of the  
footprint (and is not added to the incremental  
pixel attributes), by utilization of a set of  
four un-normalized line equations (UNLEs) L1-L4  
20 for creating a coarse mapping to determine  
texel inclusion. Each of lines L1-L4 is  
represented by its own UNLE, which is used to  
determine the inclusion or exclusion of each  
texel, within the box 14, with respect to the  
25 footprint 12a; I sum the contributions of both  
whole (completely enclosed) and partial texels  
12b to form the pixel footprint attributes.

Using an un-normalized line equation to  
determine the location of any point, and thus  
30 whether a texel should be part of the pixel  
footprint, operates by having the UNLEs



Docket 35-GS-2782

determine which side of the line the point under investigation (say, the center 12b' of each texel) is on; if a perimeter point 12a' always travels around the footprint along a side-line L in a consistent direction (say, clockwise), then a point 12b' offset from the side-line L in a known direction (say, to the right of the line L) will be potentially within the pixel - testing the point 12b' with respect to all four side-lines L1-L4 will determine inclusion or exclusion.

Therefore, the first step in the UNLE process is to generate the four un-normalized line equations; I presently prefer a UNLE of the form  $L=L_0+(L_s \times \Delta s)+(L_t \times \Delta t)$ . For example, the first UNLE, given end points (s1,t1) and (s2,t2), has coefficients of  $L_0=(s2 \times t1)-(s1 \times t2)$ ,  $L_s=(t2-t1)$ , and  $L_t=(s1-s2)$ . Each of the four UNLEs thus describes an edge of pixel footprint 12a. Each texel 12b can be tested to determine whether it is inside or out; this testing is eased if the footprint of a pixel in texel space is approximated by a parallelogram 12A, where each of the 4 sides of the parallelogram is a line described by an UNLE; the best-fit parallelogram 12A is usually one in a bounding box 14 having a major diagonal 14a close to the greater diagonal, between furthest opposed corners, of footprint 12a. Then, a point (say, a particular texel center 12b') is inside the footprint parallelogram 12A if all UNLEs yield

Docket 35-GS-2782

a positive value for that point 12b'. A four-bit indicator word is associated with each point to be tested; each bit is set by the results of testing in the associated ordered

5 UNLE (e.g., the word has bits WXYZ, where W is the sign of the first, or L1, UNLE test, X is the sign of the L2/second UNLE test for that same point, Y is the sign for the L3 UNLE test, and Z the fourth UNLE test result for line L4).

10 If a particular one of the four UNLE tests is positive, one of the 4 bits associated with the point is set to 1, otherwise, a negative result generates a 0 bit; this process is done for each of the 4 line equations and those points

15 with value of (1111 binary) or (15 decimal) are said to be inside the pixel footprint. For example, a first texel center point 12b'-1 has UNLE results of (--++) or binary 0011=3 decimal and is outside the footprint 12a, while another

20 centerpoint 12b'-2 has UNLEs of (++++ ) or binary 1111, and is inside the modified footprint 12A.

The use of four UNLEs can be simplified to two UNLE equations if the pixel footprint 12a

25 is forced to be a parallelogram 12A with two sets of spaced-apart parallel sides, having two sets of parallel line equations with the same coefficients. As previously mentioned,

$$L=L_0+(L_s \Delta s)+(L_t \Delta t),$$

30 where, for a first point at (sa,ta) and another point at (sb,tb),  $\Delta s=sb-sa$  and  $\Delta t=tb-ta$ ; for

Docket 35-GS-2782

(s1,t1) and (s2,t2) this becomes  $L1 = ((s2 \times t1) - (s1 \times t2)) + ((s1 - s2) \times \Delta s) + ((t2 - t1) \times \Delta t)$ . A line L3 parallel to the above line L1 has (s1+a,t1+b) and (s2+a,t2+b), where a and b are offset constants. Thus, a parallel line would have a first term of  $L'_0 = (s2+a) \times (t1+b) - (s1+a) \times (t2+b)$ ; which is the same as  $L'_0 = L_0 - (L_s \times a) - (L_t \times b)$ ; or  $L'_0 = L_0 + k$ ; where  $k = -(L_s \times a) - (L_t \times b)$ . However,  $L'_s = L_s$ , and  $L'_t = L_t$ , so  $L' = L + \Delta$ , where  $\Delta$  is the (a,b) offset.

In step 39, under raster-like uniform incrementation in both the "s" and "t" directions, the texels 12b enclosed in the footprint 12A can be found by using the UNLEs to determine the number of texel center points 12b' enclosed by the projected pixel footprint. If a binary word containing all 1's (i.e., 1111<sub>2</sub>) is calculated for a particular texel, that texel center is enclosed and the subject texel contributes toward the pixel attributes, otherwise the texel center is outside the footprint and the texel does not contribute to the tally. The graphics processor of the CIG may work from left to right, along each texel row (i.e., with changing s values passing through the entire required range) before incrementation of row number (i.e., change in the t value), calculating the new UNLE-based results for each of the texels: start in the upper left-hand corner of the bounding box 14 in texture space, calculate the full equation

Docket 35-GS-2782

for each of the 4 UNLE to determine each texel  
centerpoint location relative to the footprint;  
the value of the next texel centerpoint to the  
right is calculated by saving the previous  
5 value for each equation and incrementing saved  
values by their associated  $\Delta s$ ; this  
incrementation process continues across the  
row; and the same subprocess is performed for  
each succeeding row, with incrementation of the  
10 associated  $\Delta t$  amounts, until all texels within  
the bounding box 14 are considered. The  
graphics attributes (e.g., number) of texels  
having their centerpoints 12b' enclosed in the  
footprint are processed (e.g., added) to obtain  
15 the textured pixel value.

Step 41 totals the center-enclosed texel  
contributions of all texels 12b that achieve a  
1111-word result from the line equations, i.e.,  
are turned "on", with a color/ intensity  
20 contribution of 1, or else that texel is turned  
"off", with 0 (no) color/intensity contribution  
to the pixel total. It will be seen that the  
texturing effects of all texels having centers  
within the total area of the pixel projection  
25 footprint, are contributing to the pixel  
texturing, whether the projected pixel has a  
high aspect ratio (i.e., one of the width or  
length footprint dimension much less than the  
other dimension) or a low aspect ratio (i.e.,  
30 substantially equal width and length of  
footprint 12a); this process will differentiate

Docket 35-GS-2782

between a polygon at an angle of about  $\tan^{-1}$   
(2T), where T is the minimum number of texels  
in the LOD selected for the maximum axis of the  
footprint (the minimum critical dimension in  
5 the minor axial direction is the distance from  
edge to center, or about one-half the texel  
dimension, so that e.g., for an LOD yielding  
about 5 texels along the major axis of a  
particular footprint of an originally-square  
10 pixel, the aspect ratio will be about  $\tan^{-1}$  (10)  
or about  $84^\circ$ ).

Step 43 properly processes the step 41  
total, as by averaging and the like chosen  
methodology, to set the pixel texture  
15 properties. This set of properties may be  
modified, in step 45, dependent upon any other  
factors present, such as differing pixel  
lighting effects and the like. The anisotropic  
textured pixel value for that pixel is stored,  
20 in the CIG frame buffer or a separate memory,  
in step 47. Step 49 is entered and a  
determination is made of the number of pixels  
remaining to be textured for that polygon N; if  
there are additional pixels, exit 49a is taken  
25 and step 25 reentered, whereupon the pixel  
number  $p_1$  is incremented; if no additional  
pixels exist for the present polygon, exit 49b  
goes to step 51. In step 51 the same sort of  
determination is made for additional polygons  
30 in the present scene; if additional polygon  
work is to be done, exit 51a returns the

*Docket 35-GS-2782*

process to step 23; if not, exit 51b goes to step 53, which ends the texture process and passes the process to the next operation.

Referring now to Figure 2e, to improve the quality of the texturing process, each texel 12b may be subdivided into a plurality of subtexels 16; preferably, the division is into an equal number  $s$  of subdivisions along each of the S and T directions of each texel 12b, to generate  $s^2$  subtexels 16, where  $s=2,3,4,\dots$  in the presently-considered texel 12b. Just as each texel 12b has its own centerpoint 12b', each subtexel 16 has its own centerpoint 16', used here for determination as to whether that subtexel is inside or outside the footprint 12a. Thus, I prefer to subdivide each texel into  $s^2$  subtexels, where preferably  $s=2,3$  or 4; if  $s=2$ , the texel is divided into 4 subtexels 16a-16d, while if  $s=3$ , there are 9 subtexels 16a-16i, and if  $s=4$ , there are 16 subtexels 16a-16p. It should be understood that  $s$  can be any integer number of subdivisions along each side of the texel, although a square subarray within the range of  $2 \leq s \leq 4$  generally maximizes detail without unduly minimizing processing throughput (due to excessively long processing time for very large numbers of subtexels). It will also be seen that the maximum aspect angle of the underlying polygon can be almost  $89^\circ$  for a  $4 \times 4$  subtexel array, using an LOD generating 5 texels in the major texture plane direction; a

Docket 35-GS-2782

slightly greater aspect ratio, as provided by viewing the polygon 12 substantially in its plane (for an angle substantially equal to 90° with respect to the normal to the polygon plane, will cause the polygon to generate a substantially zero contribution to the displayed scenic image, which is desired, as a 90° aspect ratio causes the polygon to be seen "edge-on" and essentially disappear - no contribution should occur).

Each texel is weighted by the number of subtexels that are turned on, responsive to the same type of comparison against the signs of the four UNLEs. There will now be  $s^2$  comparisons per pixel and  $s^2$  sets of four-bit comparison words (WXYZ) to consider; if the comparison for any one subtexel is other than (1111), then that subtexel is not enclosed in the pixel footprint 12a and is not considered. Only if the subtexel comparison word is (1111) will the subtexel value be enclosed and counted in the pixel tally. Thus, for example in a 2x2 subdivided texel, if 1 subtexel is "on" (i.e., enclosed and to-be-counted), the texel value is 1/4; if 2 subtexels are on, the texel value is 1/2; while if 3 subtexels are on, the texel value is 3/4; and with all 4 subtexels being on, the value is, of course, 1 full texel's worth. This comparison is done in a modified step 39. Based on the number of subtexels enclosed in the footprint, the texel is assigned a weight that

*Docket 35-GS-2782*

contributes to the texture color of the pixel being textured.

As previously mentioned, the LOD is preferably adjusted to provide a predetermined number of sequential texels along the major axis of the projected pixel footprint 12a. Occasionally, the bounding box main diagonal 14a decreases in size to be less than one texel length, even in the finest LOD, and an optional clamp may be invoked. This clamp will maintain the shape of the pixel footprint and its centroid. The pixel footprint is then sized (i.e., magnified) so that footprint 12a fills this clamped bounding box 14. Typically, this will happen when the viewpoint 10 moves very close to the texture polygon 12a and will keep the texture from becoming very blocky. Thus, if the maximum extent of the bounding box is smaller than a texel side in the finest LOD, magnify both the footprint and the bounding box until the maximum extent is the size of the texel, so as to clamp the pixel footprint to a minimum size of one texel and anti-alias the texel in close-in views.

Referring now to Figure 3, while this method can be carried out on a programmed general-purpose computer, which acts as the graphics processor, a dedicated computational machine 60 or ASIC may be equally as well provided to carry out the process and



*Docket 33-GS-2782*

computations. Calculating the coefficients for each UNLE from 2 points requires the use of 2 digital multipliers and 3 digital adders. The precision of the arithmetic is a function of the grid size; integer arithmetic can be used.

The screen space pixel location  $P_1$ , is provided at input means 60a, for passage both to an input 62a of a pixel-to-texture mapping means 62 and to an input 64a of a pixel gradient computational means 64. Means 62 has another set of inputs 62b receiving from input 60b data characterizing the viewline 10a (e.g., angles with respect to an observation space frame) so that a set of outputs 62c can provide data establishing the texture space footprint point  $P_{tc}$ . The data at output 62c is coupled to a first set 66a of inputs to a first adder means 66, which has a second set 66b of four inputs receiving the four texel space gradients from the outputs 64b of means 64; means 64 may also receive the input 60b data at an input 64c, if required. The four adder means outputs 66c have the (s,t) coordinates of the four pixel footprint points p1-p4, respectively. From these footprint points, available to inputs 68a, a maximum extents subprocessor means 68 calculates  $S_{max}$  and  $T_{max}$  and the like, to determine the bounding box 14 and its major direction dimension 14a, which data are passed through output 68b to a first input 70a of a divider means 70. The LOD grid

*Docket 35-GS-2782*

size information provided at input 70b is used as the divisor of major dimension values at input 70a, so that the number of texels covered along the major direction of the projected pixel footprint is provide at output 70c. This number is used as the address, at input 72a, of a log<sub>2</sub> look-up-table means 72, which selects the LOD to be used, at output 72b, to an address input 73a of the graphics accelerator texture memory means 73. Means 73 outputs, at bus 73b, texel/subtexel content/attribute data.

Having determined the proper LOD, the bounding box corner point information at outputs 66c is provided to a set of adders 74 and multipliers 76 configured to provide partial terms of the UNLEs. Their outputs are fed to another adder means 78 to calculate the four UNLEs. These are provided to a first set of inputs 80a of a comparator means 80 and are compared to the texel/subtexel data from texture memory means 73 output 73b, appearing at another set of comparator means inputs 80b; if the texel 12b has been divided up into Q subtexels 16, each of the Q different subtexels (here, a 2x2 array of Q=4 subtexels) are each compared to the UNLEs by a different one of a set of a comparison means here a quad of means 80-1 through 80-4. If the subtexel 16x, where x=a, b, c or d, respectively, UNLE word is 15, then the color attribute data for that subtexel is entered into the subtexel accumulator means

*Docket 35-GS-2782*

82; after all Q subtexels are considered, the accumulator means output 82a has data for that one texel then being processed. This data is passed to the 'sum all texels' accumulator  
5 means 84, which provides the total texel contributonal data for the pixel presently being projected, to output 60b. The pixel data at output 60b can be further processed or entered directly into storage in a frame buffer  
10 and the like, for raster (or other) scanning to the CIG display means and viewing by observer 10.

While one presently preferred embodiment of my novel methods and apparatus for  
15 anisotropic texture mapping by silhouette analysis has been described in detail herein, many variations and modifications will now become apparent to those skilled in the art. It is my intention to therefore be limited only by  
20 the scope of the appending claims, and not by way of details or instrumentalities set forth herein by way of explanation.

*Docket 35-GS-2782*

What I claim is:

1. An anisotropic mapping method for determining the textural attributes of each of plural pixels in a scene to be displayed by a computer image generator, including the steps of:

(a) for each pixel, projecting a display space silhouette of that pixel as a footprint onto at least one associated object polygon in texture space;

(b) adjusting a level-of-detail (LOD) of the texels texturing the at least one polygon, to cause that projected pixel footprint to cover at least a predetermined number of texels in a selected direction of texture space;

(c) determining the accumulated number of full and partial texels covering the total anisotropic area of that footprint;

(d) determining the visual attributes for that pixel from the total texels accumulated therefore; and

(e) storing the determined visual attributes information for that pixel, to facilitate pixel display in its associated scene.

2. The method of claim 1, wherein step

(a) includes the substeps of: (a1) constructing the pixel footprint as a quadrilateral on an array of texels covering the at least one polygon; (a2) constructing a box in texture

*Docket 35-GS-2782*

space bounding the footprint; and (a3) determining a bounding box major direction.

3. The method of claim 2, wherein substep (a3) includes the step of aligning the box major direction with the further-apart pair of opposed corners of the footprint.

4. The method of claim 3, wherein substep (a2) includes the step of approximating the footprint by a parallelogram.

5. The method of claim 4, wherein the parallelogram diagonal closest to the further-apart footprint diagonal is selected as the box major direction.

6. The method of claim 1, wherein step (c) includes the substeps of: (c1) constructing a set of un-normalized line equations (UNLEs) to characterize the footprint in texture space;  
5 (c2) determining, by use of the set of UNLEs, the direction of a center of each texel within the bounding box; and (c3) including in the footprint total only those texels having a center with a predetermined relationship to all  
10 UNLEs of the set.

7. The method of claim 6, wherein a set of gradients for two-dimensional transformation

*Docket 35-GS-2782*

of the image space into the texture space, are used for construction of the UNLEs.

8. The method of claim 7, wherein step (a) includes the substeps of: (a1) constructing the pixel footprint as a quadrilateral on an array of texels covering the at least one polygon; (a2) constructing a box in texture space bounding the footprint; and (a3) determining a bounding box major direction.

9. The method of claim 8, wherein substep (a3) includes the step of aligning the box major direction with the further-apart pair of opposed corners of the footprint.

10. The method of claim 9, wherein substep (a2) includes the step of approximating the footprint by a parallelogram.

11. The method of claim 10, wherein the parallelogram diagonal closest to the further-apart footprint diagonal is selected as the box major direction.

12. The method of claim 6, further included the steps of: traversing the UNLEs in a preselected direction about the perimeter of the footprint; and including a texel in the total area only if the UNLE results for that texel are all of like selected result.

*Docket 35-GS-2782*

13. The method of claim 1, wherein each texel is divided into a plurality of subtexels, and all of steps (b)-(d) are carried out for each subtexel.

14. The method of claim 13, wherein step (b) further including the step of selecting the LOD to cause at least 5 texels to cover the pixel footprint in the selected direction.

15. A computer graphics acceleration method for selecting texel data, organized into plural levels-of-details (LODs), for use in display of image space pixels, comprising the steps of:

5       defining a scene by objects visible therein to an observer, each object being defined by at least one polygon appearing in texture space and overlaid by a collection of  
10       texels;

      defining a footprint of a display pixel silhouette projected onto the textured object;

      defining a bounding box to cover the entire area of the projected pixel footprint;

15       selecting a LOD for the texels, to cause a desired number of texels to lie along a main diagonal of the bounding box; and

      determining the total of all texels within the bounding box having a selected geometric  
20       location lying within the footprint and

*Docket 33-GS-2782*

contributing to the displayable attributes of the pixel.

16. The method of claim 15, wherein each texel is subdivided into a plurality of subtexels, and further including the step of determining subpixels as contributing to the visible attributes of the pixel if the selected  
5 geometric location of a subtexel lies within the footprint.

17. The method of claim 16, wherein the selected geometric location is the center of the subtexel being tested for inclusion in the contributing set.

18. The method of claim 15, wherein the selected geometric location is the center of the texel being tested for inclusion in the contributing set.

19. The method of claim 15, wherein plural un-normalized line equations are compared to the selected location to determine inclusion of the associated texel in the  
5 contributing set.

20. Apparatus for anisotropic mapping to determine textural attributes of each of plural pixels in a scene to be displayed by a computer image generator, comprising:



*Docket 35-GS-2782*

- 5 means, receiving a designation of that image-space pixel ( $P_{ij}$ ) then being processed, for mapping an associated pixel footprint onto at least one associated object polygon in texture space;
- 10 means for adjusting a level-of-detail (LOD) of the texels texturing said at least one polygon to be covered by at least a predetermined number of texels in a selected direction of texture space;
- 15 means for determining the accumulated number of full and partial texels covering the total anisotropic area of that footprint;
- means for determining the visual attributes for said pixel from the total texels accumulated therefore; and
- 20 means for storing the determined visual attributes information for that pixel, to facilitate pixel display in its associated scene.

ARRY C. LEE

35652782

5/15/184

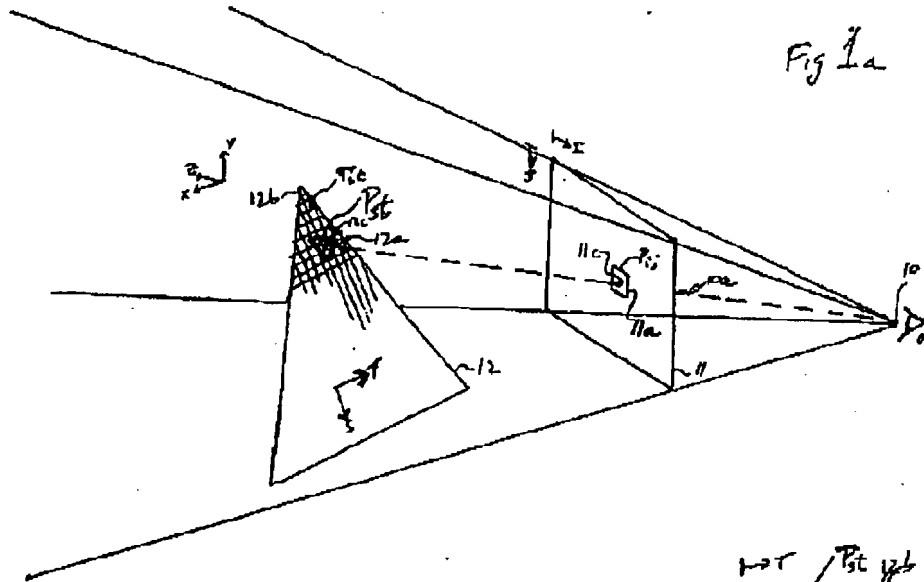


Fig 1a

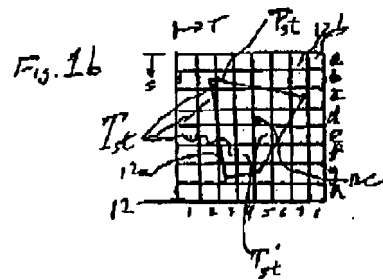
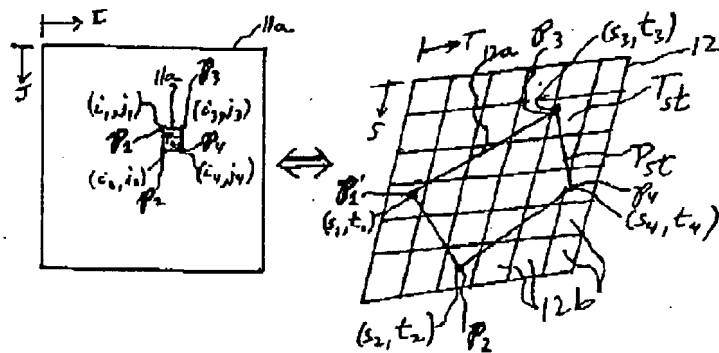


Fig. 16

Fig 2a



RY C. LEE

35652782

56267

Fig 2b

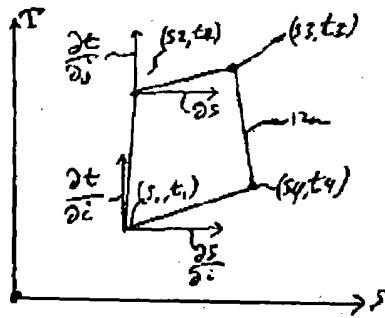


Fig 2c

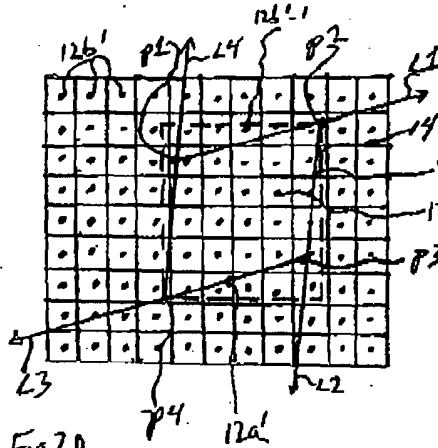
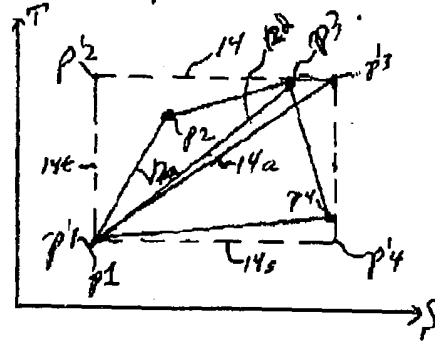


Fig 2d

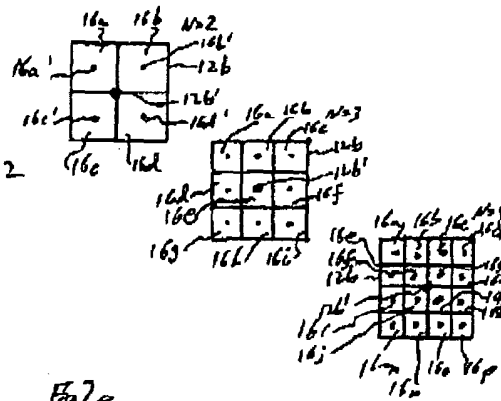
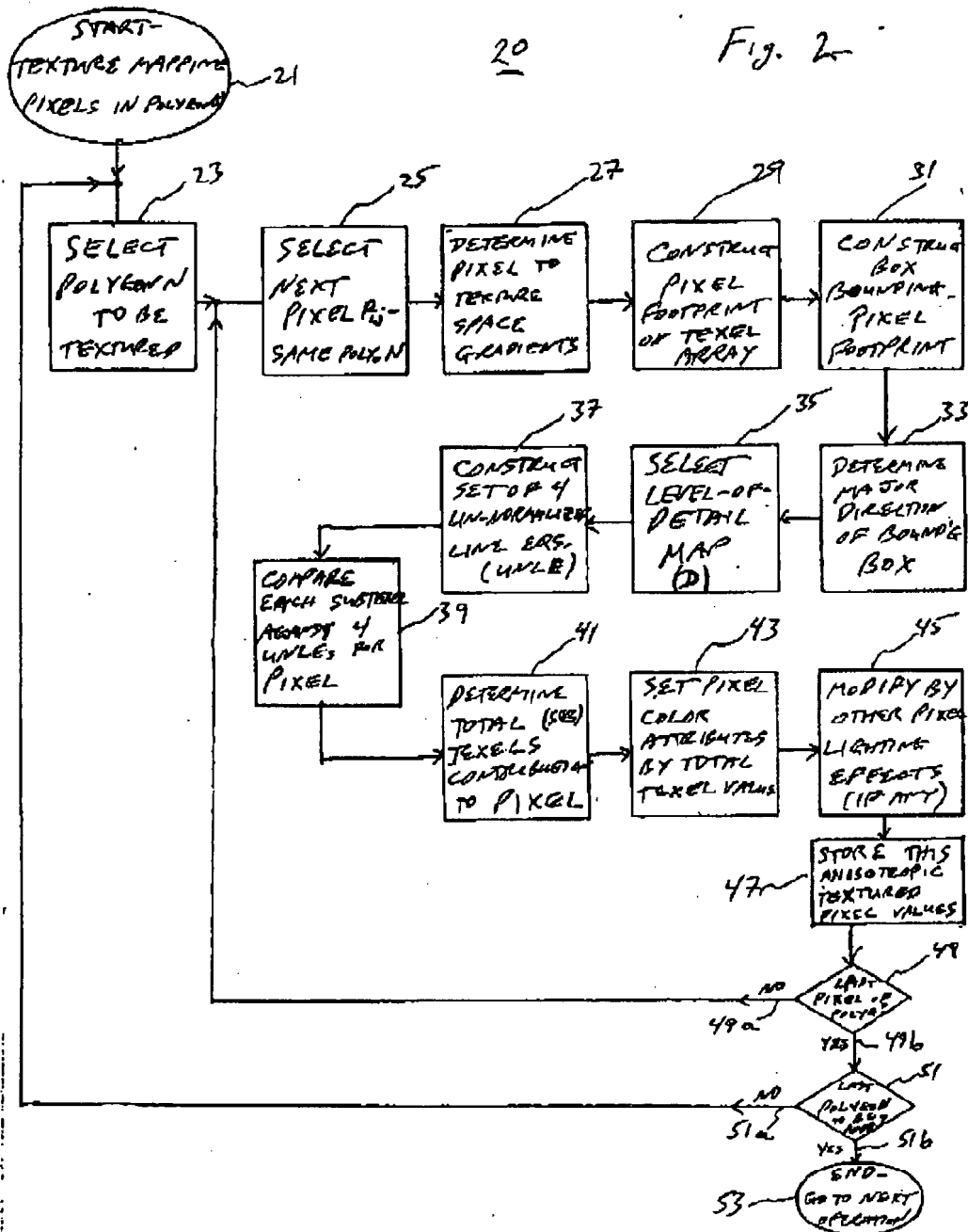


Fig 2e

SLT 3 of 4

20

Fig. 2



*Docket 35-GS-2782*

ANISOTROPIC TEXTURE MAPPING  
USING SILHOUETTE/FOOTPRINT ANALYSIS  
IN A COMPUTER IMAGE GENERATION SYSTEM

5     Abstract of the Invention

          Anisotropically mapping of texture  
information to each pixel required for display  
of each of plural polygons forming each scene  
on a CIG display in a computational efficient  
10    manner, uses the footprint of each transformed  
pixel silhouette, as mapped from image space  
into the texture array, to determine an  
anisotropic texture space area having both a  
related length and a related width, in which  
15    complete texels and texels fractions are  
covered by that pixel footprint. A set of un-  
normalized line equations characterizing a  
particular pixel footprint is used to determine  
whether each texel or subtexel in the vicinity  
20    of that pixel footprint is either enclosed or  
avoided by (i.e. inside or outside of) that  
projected pixel footprint.